Implementing Privacy-Preserving Infrastructure to Verify Personhood Credentials

1. Background

With the rapid rise in AI-generated activity, it's anticipated that humans will soon become the minority presence online. Furthermore, while anonymity has long been a core value of the internet, it has also enabled bad actors to exploit fake or misleading identities for fraud, disinformation, and other deceptive practices. As AI systems increasingly mimic human behavior—solving CAPTCHAs or generating lifelike videos—traditional identity verification methods are losing their effectiveness. To safeguard human users in a future where AI may dominate digital spaces, we need innovative solutions to reliably distinguish real people online.

2. Project Description

This project seeks to develop privacy-preserving technology that allows individuals to prove their humanity without exposing personally identifiable information. The primary objective is to implement a decentralized identity infrastructure (complemented by a front-end infrastructure developed concurrently by external, related teams) that provides backend support for securely issuing, storing, and managing verifiable personhood credentials. Key components of this infrastructure include:

1. Decentralized credential storage: Implement a simplified decentralized storage method for securely storing user credentials.
2. Credential issuance and verification: Develop a secure and privacy-preserving protocol for issuing and verifying credentials.
3. API development for external integration: Create a minimal set of secure APIs (e.g., RESTful APIs) to demonstrate backend functionality for issuing, retrieving, and verifying credentials.

2.1. Project Goals

- Develop a decentralized backend infrastructure prototype capable of securely storing and verifying user credentials.
- Implement basic privacy-preserving protocols that demonstrate secure issuance and verification of personhood credentials.
- Provide minimal API-based integration capabilities for external or frontend applications to interact with backend components.

2.2 Other Requirements

- The prototype should respond promptly to credential verification requests.
- The API endpoints should be well-documented, clearly designed, and easy to understand by other developers.
- Produce clear instructions for deployment and running the prototype (e.g., README.md file).

2.3. Ethical Requirement

All team members will be required to sign a non-disclosure agreement (NDA) upon onboarding to ensure confidentiality and responsible handling of project-related information.

3. **Desired Skill Set**
   - Programming languages such as: Kotlin, Python, Node.js, Go, or Java.
   - Decentralized storage: IPFS or similar blockchain simulator.
   - APIs: RESTful web APIs (Express.js, Flask, or similar).
   - Cryptographic libraries: Python's hashlib, Node's crypto module, or similar.

4. **Preferred Team Size**: 4 students

5. **Location of Work**: Mines Campus

6. **Primary Client Liaison**: Anhao Xiang, PhD Candidate (**Additional Collaborators:** Prof. Guannan Liu, Prof. Chuan Yue, Dr. Alemitu Bezabih, Prof. Estelle Smith)