De-SyRIS: A Decentralized Sybil-Resistant Identity System Using Mutual Facial Biometric Verification and Social Graph Analysis

Michal Ľaš
Brno University of Technology
Brno, Czech Republic
ilas.m@fit.vut.cz

Ivan Homoliak
Brno University of Technology
Brno, Czech Republic
Slovak Technical University
Bratislava, Slovak Republic
homoliak@fit.vut.cz

Juraj Mariani
Brno University of Technology
Brno, Czech Republic
imariani@fit.vut.cz

Abstract

While decentralized identity management systems strive for autonomy, most still rely on centralized mechanisms (e.g., phone numbers or government-issued IDs) to achieve Sybil resistance and prevent fake identity attacks. This article proposes a novel approach for decentralized identity management that ensures robust Sybil resistance, unlike the systems currently in widespread use. Our approach is based on a social graph of connected users. The main novelty resides in pairwise mutual facial biometric verification for connected users and anti-Sybil social graph analysis. By leveraging blockchain technology, zero-knowledge proofs, and privacy-preserving smart contract platforms, the system ensures the uniqueness, anonymity, and privacy of registered users without relying on centralized authorities. We implemented a proof-of-concept prototype of our approach using fast Plonky2 zero-knowledge proofs and evaluated its functionality through simulations on heterogeneous social graph models with 1000 entities. To the best of our knowledge, this is the first work to propose a Sybil-resistant identity system that neither sacrifices decentralization nor privacy and does not rely on any centralized credential issuers.

1 Introduction

Digital identities constitute a fundamental aspect of contemporary online life. Nearly every individual possesses some form of digital identity, encompassing online profiles (e.g., social networks), electronic IDs (e.g., OpenID Connect [30], EUID [13]), or decentralized identities [38]. Typically, these identities are managed by centralized authorities, which introduces significant privacy, security, and censorship risks [16, 19]. Self-sovereign identity (SSI) [1, 27] approaches have emerged as a decentralized option for identity management, allowing users full control over their identity-related data. However, they often rely on centralized credential issuance. For instance, to verify identity uniqueness, most systems rely on government-issued documents or phone numbers. [14, 15, 41]. Consequently, reliance on a trusted third party means the fundamental trust mechanism remains centralized, even within seemingly decentralized ecosystems. Ensuring uniqueness is crucial for preventing Sybil attacks [11], where malicious actors create numerous fake identities to disrupt a network. This article examines various design options and proposes a fully decentralized system to achieve Sybil resistance.

Contributions. We propose **De-SyRIS**, a decentralized, Sybil-resistant identity system that enables users to independently prove their uniqueness without relying on a centralized authority. The main contributions are:

(1) We integrate facial biometric verification with anti-Sybil social graph analysis. This combination establishes a layer of physical uniqueness through facial biometrics, while the social graph analysis offers a defense against sophisticated biometric spoofing and isolated Sybil clusters.
(2) We utilize a decentralized privacy-preserving smart contract platform, equipped with Trusted Execution Environments (TEEs), to process biometric data confidentially, ensuring that raw data is never exposed.
(3) We employ recursive Zero-Knowledge Proofs (ZKPs) to allow users to demonstrate their connection to a trusted social network without revealing the network's structure or their specific relationships.

2 Background

To provide a clearer understanding of the proposed identity management system, it is essential to establish the context of decentralized infrastructure and the evolution of identity management paradigms. This section provides an overview of foundational technologies and key concepts related to Sybil resistance.

Blockchain and Smart Contracts. A blockchain [3, 33, 46] is a decentralized ledger that combines cryptography, peer-to-peer networking, databases, and game theory to create a secure, transparent system for storing data. It consists of a distributed network of nodes, each maintaining a copy of the ledger. Transactions are grouped into blocks, linked cryptographically through hash functions, making the system tamper-evident. By eliminating centralized authorities, blockchain relies on consensus mechanisms, such as proof-of-work or proof-of-stake, to validate transactions. Key features of blockchain technology are decentralization, transparency, immutability, and security.

Additionally, modern blockchains support smart contracts [39], which are self-executing programs that enforce rules autonomously, eliminating the need for intermediaries.

Identity Management Systems. Identity Management Systems (IDMSs) [16, 19] manage the lifecycle of digital identities, including registration, verification, and revocation. These systems are conventionally categorized by the architectural centralization of the Identity Provider (IdP) as *centralized*, *federated*, and *decentralized*. Centralized and federated systems rely on a single authority for user data management, which might cause issues like single points

of failure, potential spoofing, and censorship. In contrast, decentralized systems emphasize Self-Sovereign Identity (SSI), empowering users to manage their own identity data.

Decentralized Identifiers and Verifiable Credentials. Decentralized identities [38] (DIDs) and Verifiable Credentials [37] (VCs), standardized by the World Wide Web Consortium (W3C), represent the latest trend in identity management.

A DID is a globally unique URI that resolves to a DID document containing public keys and authentication materials, uncoupled from any centralized registry. The Verifiable Data Registry (VDR), typically a blockchain, hosts these DID documents.

Verifiable Credentials provide a mechanism for expressing claims about a subject. The VC ecosystem involves three primary actors: the *issuer* (who cryptographically signs the claims), the *holder* (the user who stores the VC in a digital wallet), and the *verifier* (who requests cryptographic proof of the claims). These architectures often enable selective disclosure and zero-knowledge presentations of claims to enhance user privacy.

Zero-Knowledge Proofs. A Zero-Knowledge Proof [4, 25] is a cryptographic protocol involving two parties: a *prover* and a *verifier*. The protocol enables the prover to convince the verifier that they know a specific piece of information without revealing the information itself. A robust ZKP protocol must satisfy three fundamental properties: *Zero-Knowledge*, *Completeness*, and *Soundness*. The proposed system utilizes **Succinct Non-Interactive Arguments of Knowledge (ZK-SNARKs)**, which are efficient, small, non-interactive proofs that enable fast verification.

Recursive zk-SNARKs. A zk-SNARK circuit can incorporate the verification logic of another zk-SNARK proof. Each zk-SNARK verification can be viewed as a computation represented by a circuit that can be used in another zk-SNARK. This feature allows combining multiple proofs into a single proof, confirming the correct execution of the entire sequence.

Trusted Execution Environments. A Trusted Execution Environment (TEE) is a secure, hardware-isolated area within a processor. It guarantees the confidentiality and integrity of the code and data loaded inside it. Even the hardware owner or the host operating system cannot inspect or alter the execution state within the TEE.

Privacy-Preserving Smart Contract Platforms. There are various decentralized solutions that provide verifiable, confidential execution [2, 28, 44, 47]. These systems integrate TEEs, such as Intel Software Guard Extensions (SGX) [9] into their execution layer to achieve private execution.

3 Problem Definition

Many services necessitate unique digital identities to ensure that each physical person is represented **exactly once** within the system. The problem we tackle is that even in decentralized identity management systems, Sybil resistance is achieved almost exclusively through centralized solutions. These solutions [13, 15, 30, 41] rely on verifying centralized government-issued IDs or passports that contain biometric and demographic information. Digital identities are also managed by social networks and other institutions, such as banks. While social networks use social graph or behavioral analysis of users, other institutions may use entity resolution [42] to ensure Sybil resistance. As a result, users cannot prove that their virtual identity is legitimately linked to their real-life identity without dependence on a centralized authority.

Desired Properties. A robust Sybil-resistant identity system must satisfy several key properties:

- ***Uniqueness:*** Ability to ensure one virtual identity per entity.
- ***Humanness:*** Verification that registered entities are living humans.
- ***Privacy, Anonymity, Unlinkability:*** Maintaining control over personal information while ensuring unlinkability, which stops attackers from connecting virtual identities to real identities.
- ***Decentralization:*** Absence of a single point of failure or control.
- ***Key-Recovery & Revocation:*** Ability to recover access if keys are lost or revoke keys if stolen.

Meeting all of these properties is challenging. Existing decentralized solutions often involve trade-offs among decentralization, privacy, and usability. For example, two widely used decentralized solutions are World Network [45] and BrightID [6]. WorldNetwork uses iris scan-based registration. However, this approach requires specialized hardware, whose utilization for registration is centralized. BrightID uses a social graph analysis approach based on various algorithms. Entities are pseudonymized, but their connections are public. Simply pseudonymizing entities is insufficient; if one person's pseudonym is disclosed, it can lead to the gradual revelation of other identities based on their connections, ultimately compromising the anonymity and connections of the entire social graph. Furthermore, social connections themselves are considered sensitive information. In this work, we propose a fully decentralized, Sybil-resistant identity system that eliminates the need for trusted credential issuers while maintaining both privacy and usability.

4 System Architecture Overview

De-SyRIS integrates facial biometrics with social graph analysis to establish uniqueness and humanness without reliance on a centralized authority. No special hardware is required to perform a facial scan; users can do so using their smartphones. The processing and comparison of registered facial biometric data occur within the TEE to protect the privacy of biometric information.

Social Graph Analysis. Social graph analysis uses a Privacy-Preserving Path Discovery (PPPD) (see Section 5.3.2) and a route-based defense mechanism. This mechanism assumes that, while an attacker can easily forge numerous virtual identities and create arbitrary connections among them, establishing connections with honest users is exceedingly difficult. This creates a distinct topological separation: an **honest region** and a **densely connected Sybil region**, separated by a sparse set of attack edges.

Bootstrapping & Principle of Operation. The verification process is fundamentally divided into three sequential phases:

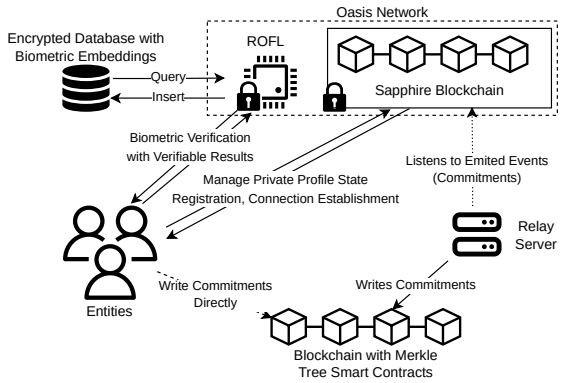

Figure 1: Actors of the proposed Sybil-resistant system and their interactions.

(1) **Registration:** An entity enrolls their facial biometrics using a decentralized privacy-preserving platform. This establishes a unique, privacy-preserving profile on the blockchain without publicly exposing sensitive biometric data.
(2) **Connection Establishment:** Registered entities form mutual, private connections by physically meeting and verifying each other's biometric uniqueness utilizing the stored template from step (1) and provided identity information. These mutual verifications construct the edges of the decentralized social graph. Entities that establish connections must trust one another because they share sensitive data. Connections typically include family, friends, and acquaintances, not strangers. The system protects biometric data and connection details from the wider network, not established connections.
(3) **Proof of Connection with Honest Entities (Path Discovery):** To demonstrate integration into the honest region of the network and prove Sybil resistance, entities generate recursive ZKPs of their connections. This process, called *Privacy-Preserving Path Discovery* (PPPD), allows entities to construct continuous paths through the graph without ever revealing the identities of connected entities or the graph's topology to the public.

To facilitate these processes while guaranteeing privacy, the system employs the Oasis Network [28, 29], which is a **decentralized**, privacy-preserving smart contract platform and a L1 blockchain. The primary system actors and their interactions are illustrated in Figure 1, and they consist of:

- **Entities:** Users operating the system through mobile devices equipped with trusted hardware for generating authenticated face scans.
- **Oasis Sapphire:** A confidential smart contract layer that maintains the encrypted state of entity profiles and connections. The privacy-preserving capabilities ensure that social graphs are never exposed publicly.
- **Oasis ROFL:** A stateless, verifiable compute layer executing heavy tasks such as facial biometric processing and comparison. Execution produces a result and an attestation verifiable on-chain.
- **Database:** A centralized[1] but encrypted repository storing biometric embeddings. It is accessible exclusively by ROFL enclaves, with authentication managed via Remote Attestation quotes.
- **Relay Server:** An infrastructure element that listens for events emitted by Sapphire and batches commitments into an on-chain Merkle tree.

Complementary Synergy of Biometrics and Social Graphs. While facial biometrics provide a strong foundational identity, they are prone to false positives and sophisticated physical forgeries, especially when scans are executed on consumer devices. The social graph analysis effectively neutralizes these shortcomings. Even if an attacker successfully circumvents biometric verification to register multiple synthetic profiles, establishing the requisite mutual real-world connections with honest entities to integrate those profiles into the trusted social graph remains difficult. Conversely, the biometric requirement increases the cost and complexity of creating attack edges (social connections) in the social graph. Although it is theoretically possible to create numerous identities using fake biometric data, establishing a connection requires mutual biometric verification. The resulting social graph will be relatively conservative. Entities will likely not connect with just anyone, since they would have to provide a face scan and personal data.

5 Details of De-SyRIS

This section elaborates on the individual processes introduced in the overview, detailing the cryptographic and architectural mechanisms for registration, connection establishment, and social graph analysis.

Established Notation. Throughout the system, it is assumed that the entity's primary identifier (ID) is their wallet address, which is deterministically derived from their public key (PK). Furthermore, to formalize the cryptographic processes, the following notation is established: (1) ID_A: The unique identifier of the entity A (derived from public key PK_A); (2) SK_A: The private key of entity A; (3) FS_A: The facial scan of entity A; (4) E_A: The multidimensional biometric embedding extracted from FS_A; It can have the upper index s for *stored*, or c for *captured*; (5) $H(E_A)$: The hash of the normalized embedding E_A; (6) t_A: A timestamp generated by the ROFL enclave; (7) CC_{AB}: The connection commitment between A and B; (8) S_{AB}^{CC}: A shared secret salt used to blind the connection commitment; (9) CC_{AB}^{s}: A salted connection commitment between A and B; (10) $Sig_A\{content\}$: Digital signature made by entity A with their SK_A. Note that the lower index always indicates the entity or entities to which a given parameter belongs.

5.1 Registration

The registration process enrolls an entity's facial biometrics, thereby establishing a unique profile on the blockchain. The entire process is illustrated as a sequential diagram in Figure 2. The verification

[1]Although this setup may seem like a single point of failure, the database can be decentralized or at least replicated across multiple instances. However, the performance and reliability of decentralized databases are limiting factors.

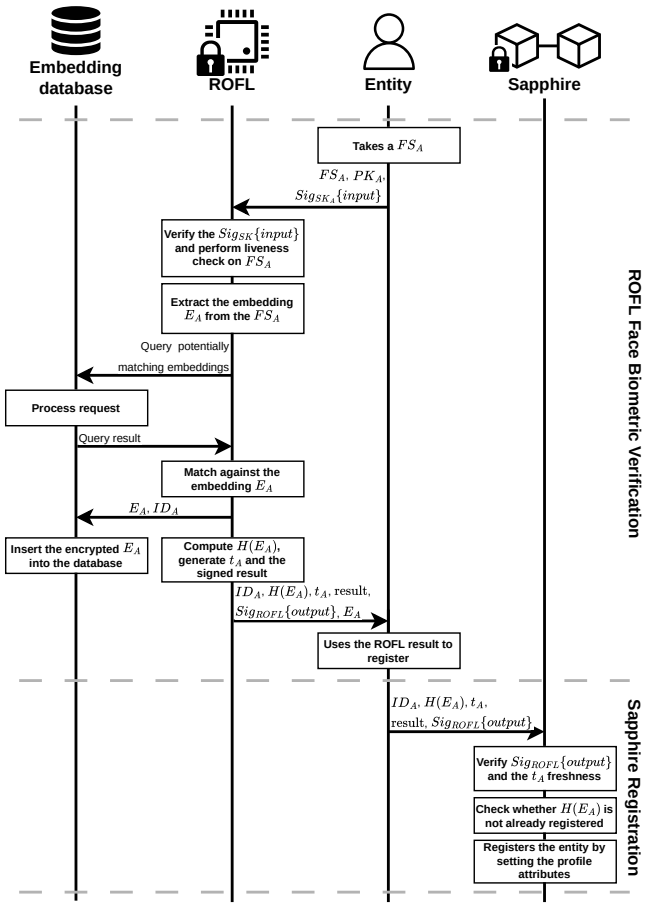

Figure 2: De-SyRIS profile registration process.

occurs entirely within TEEs to preserve privacy and prevent unauthorized access to biometric data. The procedure is formalized in two distinct stages: **ROFL Face Biometric Verification** and **Sapphire Profile Registration**.

5.1.1 ***ROFL Face Biometric Verification.*** During face biometric verification, the entity's A device captures a complete facial scan (FS_A) and sends it to ROFL for verification. The process can be summarized as follows:

- **Input:** FS_A, PK_A, and $Sig_{SK_A}\{input\}$.
- **Output:** ID_A, $H(E_A)$, t_A, match result, and the output signature $Sig_{ROFL}\{output\}$. Additionally, ROFL can return E_A for local storage.

(1) Verify the $Sig_{SK_A}\{input\}$ to authenticate the entity and the integrity of the message.
(2) Perform robust active and passive liveness checks on FS_A to ensure the scan represents a real, live human. This can involve complex liveness verification, such as interactive challenges issued by the ROFL enclave to the entity.
(3) Extract the biometric embedding E_A from the FS_A.
(4) Query the encrypted database for potentially matching embeddings.
(5) Decrypt the queried embeddings within the TEE and match them against the embedding E_A currently being registered.
(6) If no match is found, insert the encrypted E_A into the database.
(7) Compute hash $H(E_A)$ to track unique face scans and prevent repeated use of the same FS_A. Before hashing, there should be a normalization of E_A to resolve floating-point inaccuracies, ensuring that hashing the same FS_A embedding consistently yields the same result.
(8) Generate a new timestamp t_A to ensure freshness during the Sapphire registration.
(9) Return a signed result $Sig_{ROFL}\{output\}$ that includes ID_A, $H(E_A)$, t_A, and the match result. Additionally, return E_A so that the entity can store it locally.

The ROFL enclave must authenticate itself to the database using Remote Attestation quotes to prove its integrity before being granted access to query or insert encrypted embeddings.

5.1.2 ***Sapphire Profile Registration.*** The user uses the signed ROFL result $Sig_{ROFL}\{output\}$, including ID_A, $H(E_A)$, t_A, and the match result, to register a new account on the Sapphire blockchain. The process can be summarized as follows:

(1) Verify $Sig_{ROFL}\{output\}$ to ensure hardware-backed verification and the freshness of the t_A.
(2) Check if $H(E_A)$ is already registered in the contract state.
(3) The smart contract registers the entity by storing the $H(E_A)$, setting the reputation score, and recording the registration state, either **fully registered** (where no biometric match was found) or **limited registration** (triggered by a biometric match of an already existing record).

In the case of limited registration, the entity must obtain community endorsements from already registered entities to achieve full status. An endorsement is a special vote that serves as a secondary, community-driven layer of Sybil resistance designed to overcome the natural false-positive rate of facial recognition. Each fully registered entity can give only one endorsement to prevent indiscriminate vouching. Once a required endorsement threshold is reached, the biometric embedding of the entity with limited registration is inserted into the database, and its status is updated to full.

5.2 Connection Establishment

Fully registered entities can establish mutual connections. These connections constitute the edges of the social graph but remain entirely hidden from the public.

The establishment protocol is executed in three steps, as illustrated in Figure 3: **In-person Meeting**, **ROFL Mutual Verification**, and **Sapphire Connection Registration**.

5.2.1 ***In-person Meeting.*** During an in-person meeting, the entities A and B exchange their wallet addresses (ID_A, ID_B) and take facial scans of each other (A takes FS_B, B takes FS_A). Additionally, A and B can exchange the stored embeddings (E_A^s, E_B^s), so they do not have to be queried from the database.

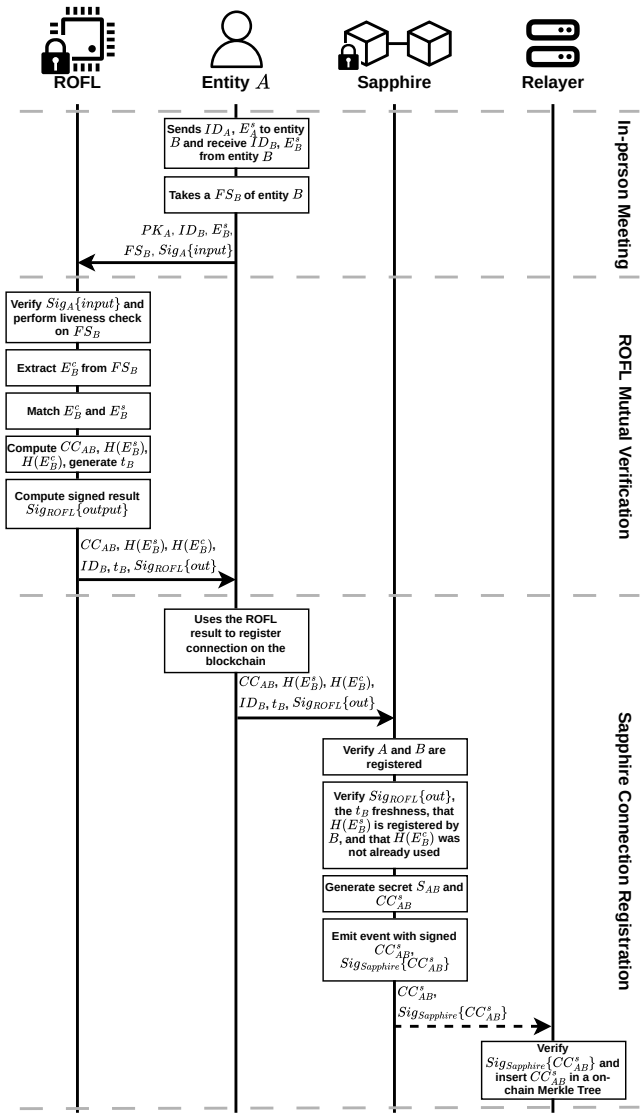

Figure 3: Connection establishment process between two entities from the perspective of Entity A.

5.2.2 ***ROFL Mutual Verification.*** After exchanging data, both entities send the signed, encrypted data to Oasis ROFL for verification. Both entities perform the same actions. Therefore, the verification protocol is described only from the perspective of A:

- **Input:** PK_A, ID_B, FS_B, E_B^s, and $Sig_A\{input\}$.
- **Output:** CC_{AB}, $H(E_B^s)$, $H(E_B^c)$, ID_B, t_B, and the output signature $Sig_{ROFL}\{output\}$.

(1) Verify the $Sig_A\{input\}$ to authenticate the entity and the integrity of the message.
(2) Perform liveness checks to ensure the facial scan is a live capture rather than a presentation attack.
(3) Extract embedding E_B^c from FS_B.
(4) Match embedding E_B^c against the provided registered embedding E_B^s. If E_B^s was not provided, ROFL can query it from the database using the ID_B.
(5) If matched, compute ID_A from PK_A and the connection commitment $CC_{AB} = H(\min(ID_A, ID_B) \parallel \max(ID_A, ID_B))$[2].
(6) Generate a new timestamp t_B to ensure freshness during the connection registration on Sapphire.
(7) Return a signed result $Sig_{ROFL}\{output\}$ including CC_{AB}, $H(E_B^s)$, $H(E_B^c)$, ID_B, and t_B.

5.2.3 ***Sapphire Connection Registration.*** Following successful ROFL verification by both parties, the connection is formalized on the blockchain via the Sapphire confidential smart contract, using the ROFL result. The process can be summarized as follows:

(1) Verify that both A and B are registered, that the signature $Sig_{ROFL}\{output\}$ is valid, that the timestamp t_B is fresh, that $H(E_B^s)$ is the registered embedding of B, that $H(E_B^c)$ was not already used, and that the provided CC_{AB} is not registered yet.
(2) Store $H(E_B^c)$ for the duration of its timestamp validity to prevent reuse.
(3) Generate a random secret S_{AB}^{CC} serving as a shared salt and the salted commitment $CC_{AB}^s = H(CC_{AB} \parallel S_{AB}^{CC})$.
(4) Emit a connection establishment event with signed CC_{AB}^s, $Sig_{Sapphire}\{CC_{AB}^s\}$.
(5) The relayer listens for these events, verifies the signature $Sig_{Sapphire}\{CC_{AB}^s\}$, batches the connection registrations, and inserts them into the public on-chain Merkle tree.

A connection can also be revoked by inserting its nullifier into a revocation Sparse Merkle Tree (SMT). In such a case, the record of the established connection on the Sapphire blockchain is also deleted.

5.3 Social Graph Analysis

Registered entities with established connections can demonstrate their membership in a broader network. This is achieved using recursive ZKPs. The two primary protocols utilized for this purpose are **ZK PPPDs** and ZK **Proof of Paths**.

5.3.1 ***Reputation.*** To distinguish an honest region from a Sybil region, we need known, trustworthy entities. The structure of the regions alone is not sufficient, since the attacker can connect their controlled Sybil entities in any way and effectively mimic the structure of the honest regions. Since we want the entities to be equal, the system relies on an external reputation system, which, however, is not part of this solution. This system assigns a reputation score to entities that take the initiative and get additional verification.

It is highly desirable for this reputation system to be decentralized. For instance, it could use decentralized verification via Decentralized Autonomous Organizations (DAOs), verification of verifiable credentials [37], or public identity verification for well-known individuals, such as politicians, academics, influencers, lawyers, doctors, etc. The simplest method would involve centralized identity verification using issued digital documents, such as passports or ID cards. However, opting for centralized methods would make

[2]This ensures that the connection commitment will be the same for both entities.

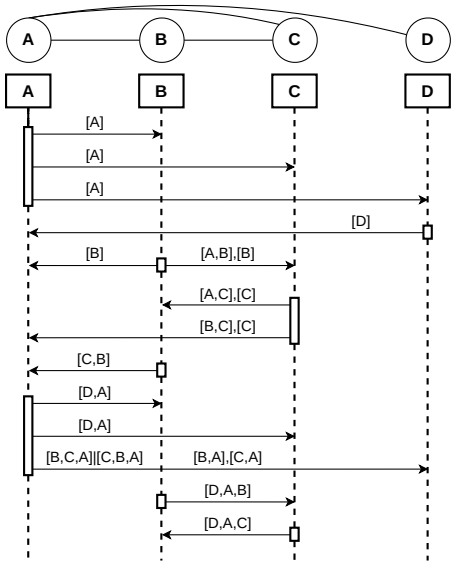

Figure 4: Step-by-step construction of a ZK PPPD among four entities. The diagram above shows how the entities are connected. For example, [A, B, C] denotes a path of length 3, starting with entity A, passing through entity B, and ending with entity C. A initiates a path and sends a proof to B. B extends this proof by itself and sends the path [A, B] to C. C can check that their connected peers A and B are already on the path and therefore does not propagate this path further. C can verify that its peers A and B are already on the path and therefore does not propagate this path further. Each entity propagates to its peers only those paths it has not yet traversed, and does so asynchronously.

the system less decentralized, so it's preferable to rely on more decentralized approaches.

Importantly, **only a small fraction of entities need this external reputation**, as the majority can be verified through transitive connections.

To encourage participation, an entity's total reputation is divided into two components: **external** reputation and **intrinsic** reputation (which is earned through protocol activity). Both components have maximum limits, and their combined total is committed to an on-chain Merkle tree, enabling verification through Zero-Knowledge Proofs (ZKPs).

5.3.2 ***ZK Privacy-Preserving Path Discovery (PPPD).*** Entities construct **paths** through the social graph by recursively proving their connections and handing these proofs to each other. This process is illustrated in Figure 4.

The proof states the following: Entity A, as the intended recipient of a valid path handed off by entity X, proves they share an unrevoked connection with X, possess a valid reputation score, and have not previously participated in the current path, while correctly updating the path's length and accumulated reputation before securely locking the new proof for their connection B.

The formula for calculating path reputation by A (prover) to B (next entity) is: $pathR_{new} = pathR_{old} \times \alpha + R_A \times w_A^B$, where $pathR$ is the path reputation (initially 0), $\alpha \in [0, 1]$ is the decay factor, R_A is the entity's reputation (external + intrinsic, both have maximum), and $w_A^B \in [0, 1]$ is the weight of the connection in the direction from A to B. The weight indicates how much trust entity A has in B, with A setting the weight for its connections.

Valid paths require a length of $O(\log n)$, where n is the number of entities in the network, and a reputation score high enough to guarantee the inclusion of at least one externally reputable entity. Paths must be without cycles, but tracking entities explicitly (e.g., $A \rightarrow B \rightarrow C \rightarrow A$) would compromise privacy by revealing network topology.

To prevent this, we propose using a **Bloom filter**. Instead of an explicit array, entities are inserted into the filter. This hides specific identities while allowing the network to check if an entity is already on the path, yielding false positives but never false negatives.

In the ZK Proof of Paths, distinctness between multiple paths is verified using a pairwise bitwise-AND across the Bloom filters. If the resulting filter, excluding the terminal entity in the path, which is always the same, has fewer than k set bits (where k is the number of hash functions, accounting for the single shared proving entity), the paths are highly likely to be distinct. For example, for paths of length 16, a 4096-bit filter with 10 hash functions yields $\approx 3.8467\%$ false positives, and a 8192-bit filter with 10 hash functions yields $\approx 0.0283\%$ false positives. Ultimately, this verifies path distinctness, preventing Sybil attackers from exploiting single bottlenecks while maintaining absolute topology privacy.

5.3.3 ***ZK Proof of Paths.*** Once an entity has gathered the required number of distinct paths (sets of entities across all paths must be disjoint), it can create a ZKP that aggregates these proofs into a single proof. The required number of paths should be $O(\log n)$, where n is the number of entities in the network. The smart contract on Sapphire verifies this proof and determines whether the account is non-Sybil. In addition, it increases the entity's intrinsic reputation based on the reputation it has gathered across all paths.

In this way, even entities without an external reputation can gradually build their own reputation scores and thus rely less on other entities.

5.4 Epochs

Reputation and established connections can change over time. For example, connections can be revoked if an entity is compromised or attempts to harm the system. However, tracking these changes poses challenges for anonymity and privacy. To address this, DeSyRIS employs an epoch-based system. Changes are implemented at the beginning of each epoch, and the same state remains in effect throughout that period. Consequently, this requires periodic ZK Proofs of Paths to ensure verification remains up to date. ZK Proofs of Paths operate without any user action. They use existing connections, so there is no need for repeated in-person meetups or mutual biometric verification. The length of a single epoch determines how up-to-date the system will be. A safe and conservative setting would be 3–7 days.

5.5 Properties of Proposed System

The proposed system addresses the key properties desired in a Sybil-resistant identity system. It ensures **uniqueness and humanness**

by combining facial biometrics with social graph analysis to prevent forgeries and mass registration. **Privacy, anonymity, and unlinkability** are heavily protected by processing biometrics within secure TEEs and using ZKPs to conceal trust connections. The entire system is **decentralized** and does not rely on a centralized authority.

Currently, the system does not natively meet the requirements for **key-recovery and revocation**. However, this can be resolved in future implementations by leveraging the decentralized social graph to enable key recovery through trusted peers rather than centralized authorities.

6 Evaluation

The evaluation is based on a proof-of-concept (PoC) implementation and a simulation of the PPPD mechanism in a social graph using various synthetic models. The performance of the ZKPs is a key aspect of the PoC implementation. The simulation was designed to verify the ability of honest entities to authenticate themselves and to demonstrate Sybil attacks and their success rates in practice.

6.1 Simulation of the Social Network Analysis

Three graph models were utilized to reflect real-world network properties:

- **Watts-Strogatz:** captures the small-world phenomenon by producing graphs with short average path lengths and high clustering. The parameters k (mean degree) = 20 and p (rewiring probability) = 0.02 were set.
- **Holme-Kim:** generates scale-free networks with high local clustering to simulate tightly-knit communities with highly connected hubs. The parameters m (edges added per new node) = 10 and p_{tri} (probability of triad formation) = 0.6 were set.
- **Kleinberg:** adds probabilistically distributed long-range links to grid networks, demonstrating how entities can efficiently find short trust paths in decentralized structures. The parameters p (local radius) = 3, q (number of long-range edges) = 2, and r (clustering exponent / distance penalty) = 2.5 were set.

6.1.1 ***Simulation of Trust Distribution.*** This simulation aimed to model the distribution of trust among 1000 entities over 5 epochs, with an increasing number of entities having external reputations. This simulation demonstrates how much external reputation is needed in the network for honest entities to get verified. Reputation scores were generated using a normal distribution parameterized so that the effective range is constrained to [0, the maximum reputation], with the mean centered on the midpoint. Similarly, edge weights are also randomly generated from a normal distribution. The results are presented in the graph in Figure 5.

The amount of network reputation that entities need for effective verification depends largely on how densely they are interconnected. In the Holme-Kim model, more entities have many connections and function as hubs connecting different parts of the network; therefore, only 10% of entities with external reputation are sufficient. The Watts-Strogatz and Kleinberg models are more conservative and demonstrate the small-world paradox. Nevertheless,

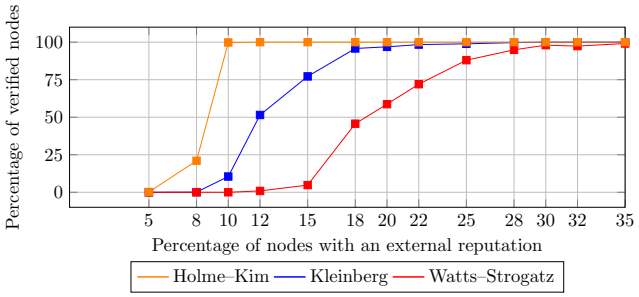

Figure 5: Results of the entity verification simulation based on the number of entities with external reputation.

Example	Circuit		Proof		
	Comp.	Size	Prove	Ver.	Size
ZK PPPD Linear	1.56s	8KB + $l \times$ 4.2MB	328.56ms	4.44ms	124KB
ZK PPPD Cyclic	2.39s	8.4MB	630.27ms	4.68ms	130KB
ZK Proof of Three Paths	1.25s	16.9MB	1.58s	7.19ms	143KB
ZK Proof of Six Paths	2.65s	33.8MB	2.82s	12.41ms	149KB

Table 1: ZK PPPDs and ZK Proof of Paths circuit compilation times, sizes, and proof generation times, verification times, and proof sizes. The l parameter denotes the length of the path. Performance was tested on an AMD Ryzen 5 5600H with 16 GB of RAM.

even in these models, the network begins to function effectively when 12–18% of entities have external reputation.

6.1.2 ***Simulation of Sybil Attacks.*** This simulation utilized the results from the first one. From each model, we selected the reputation distribution at which honest entities began to effectively verify each other: 18% for Watts–Strogatz, 10% for Holme–Kim, and 12% for Kleinberg. The simulation created a cluster of 200 Sybil entities; attack edges were gradually added to honest entities, and the number of verified Sybil entities was tracked. Sybil entities lack an external reputation, unlike honest entities, which may have one.

To verify the entire Sybil cluster, at least as many edges are needed as there are distinct paths. Furthermore, these edges must be distributed evenly among the entities, because if one entity had more edges, it would effectively be a single path for the other Sybil entities, since the path would always pass through that one Sybil entity.

However, once this number is reached, Sybil entities' ability to achieve verification depends on the distribution of reputation. Although more attack edges give Sybil entities more options, here, it matters whether the Sybil entity has connected to an already-verified entity, whether the honest entity has an external reputation, how large that reputation is, and what weight the honest entity has assigned to this connection. These values were generated randomly from a normal distribution.

Due to the number of variables, it is difficult to evaluate all cases and the actual impact. For example, in some runs of the Holme-Kim model, even 20 attack edges were not enough to verify a Sybil cluster. Typically, only Sybil entities that possessed attacking edges were verified. In contrast, in other runs, nearly the entire Sybil cluster managed to verify itself with the minimum possible number of attack edges.

6.2 ZK Proof Performance

De-SyRIS contains two recursive ZKPs, and – especially in the case of ZK PPPDs – the speed of these proofs is important because they are generated repeatedly. The Plonky 2 [31] scheme was selected for these ZKPs for its highly efficient recursion and optimized Poseidon hashing. Two circuit designs for the PPPDs were evaluated: a *linear* approach (one distinct circuit per recursion depth) and a *cyclic* approach (a single, self-referential universal circuit). Furthermore, the final ZK Proof of Paths aggregates multiple PPPDs. The results are shown in Table 1. Note that the ZK PPPDs in the PoC implementation do not use the proposed approach with Bloom filters; instead, they use a simple array of nullifiers.

7 Security Analysis

Attackers will attempt to create a large number of verified Sybil entities. To do so, they need to bypass biometric registration and establish sufficient connections to the honest part of the network. In addition, identity theft may occur, and it is important to protect privacy and anonymity and prevent tracking of users' actions, as this could lead to censorship.

7.1 Facial Biometric Fraud and Forgeries

The threat of Sybil creation via sophisticated physical masks, AI-generated deepfakes, or maliciously altered captures is a primary challenge when dealing with user-submitted biometrics. It is also possible to steal someone's facial biometrics. That is why advanced biometric verification is needed not only during registration but also during connection establishment. Since the verification relies on the user's device, the system requires a full facial scan rather than a simple 2D photograph. This raises the bar for attackers, as generating a full 3D scan deepfake in real-time or creating a high-fidelity 3D physical mask is more difficult than presenting a manipulated image. Furthermore, the ROFL enclave must execute robust active and passive liveness detection algorithms. Active liveness challenges, which prompt the user to perform randomized actions during the scan, could further strengthen this defense.

Even if an attacker successfully bypasses the liveness detection to create fake identities, these identities still face the challenge of integration into the social graph. The requirement for mutual, in-person biometric verification means the attacker would have to physically present the spoofing mechanism (e.g., a highly realistic mask) to honest users to establish connections, or remain isolated in a disconnected Sybil cluster. Thus, the combination of advanced liveness detection, full facial scans, and social-graph-based mutual verification provides a defense-in-depth approach against biometric fraud.

7.2 Attacks on Social Graph Analysis

In the designed system, for a Sybil cluster to be recognized, the attacker must obtain the necessary number of distinct attack edges (connections) to the honest region. This means that the attacker cannot gain an advantage by using a single biometric forgery to deceive multiple honest entities. Instead, the attacker will need as many forgeries as the minimum required number of distinct paths. A simulation of such an attack is described in Section 6.1.2.

However, a coordinated group of malicious actors could leverage their real, highly reputable identities as gateways to legitimize a massive Sybil cluster. Furthermore, the system is not resistant to entities that intentionally trade their facial biometric data and establish connections with untrustworthy entities, which may be Sybil accounts, thereby enabling them to obtain verification. An additional mechanism to penalize such behavior is needed. However, this is a broader problem with identity trading, which is difficult to solve in any decentralized system.

7.3 Censorship Attacks

The system architecture introduces specific vectors for censorship, categorized into relayer-based and entity-based censorship models.

Relayer Censorship. The relayer is responsible for submitting connection and reputation commitments from the Sapphire to the public blockchain. A malicious or compromised relayer could ignore commitments from targeted entities. The system mitigates this by allowing users to bypass the relayer and submit transactions directly to the smart contracts. Although direct submission sacrifices some unlinkability, mixing services can be employed to preserve privacy during this contingency.

Entities Censorship. The recursive PPPD protocol relies fundamentally on peer cooperation. Users may arbitrarily refuse to forward path proofs for specific individuals, effectively censoring them from the broader network. Given the decentralized structure and the multiplicity of connections required, isolated censorship by a few entities has a statistically negligible impact on an entity's verification process. Furthermore, the protocol incentivizes continuous cooperation through an intrinsic reputation reward mechanism. Entities that continuously participate in generating and forwarding proofs receive intrinsic reputation across epochs, thereby optimally aligning individual incentives with the overall network's operational health and security.

8 Discussion

Facial Scan False Positives. While community endorsements mitigate biometric false positives, they risk being coerced, bought, or remaining inaccessible to honest users. Consequently, utilizing high-quality facial embedding models and improving the endorsement mechanism remain critical priorities.

Trust in the Social Graph. Defining the minimum reputation threshold for a valid path involves a trade-off: setting it too high creates a *cold start* problem for new users, while setting it too low admits Sybils. Currently, based on the maximum intrinsic reputation achievable without external reputation, future work should explore dynamic thresholds that adapt to the network's evolving trust distribution.

Bootstrapping Problem. Like many existing approaches based on social graph analysis, this system may have problems with bootstrapping. It may take some time for entities to establish the necessary number of connections and obtain verification.

Persistence of Merkle Proofs. ZKPs require historical Merkle roots to remain available on-chain for verifiers. While typically addressed using a historical root buffer that lasts for an entire

epoch, managing this buffer without introducing centralized relayer dependencies remains a significant smart contract engineering challenge.

Security of TEEs. The system relies on a decentralized network of TEEs, whose viability is threatened by emerging hardware attacks. For instance, the Integretee project [23] was discontinued following the Intel SGX WireTap vulnerability [34], which Intel will not patch as it assumes physical security. Consequently, the secure functioning of TEEs in decentralized environments remains an open question.

Alternatives for biometric verification are decentralized biometric identification systems; for example, Hu et al. [21], Outsourced Biometric Identification With Privacy (TIFS 2018). However, these systems are still under active research. For confidential computing and encrypted smart contracts, it is possible to utilize existing public blockchains and fully homomorphic cryptography, such as the Fhenix protocol [17, 50].

Future Enhancements: Hardware-Backed Provenance (C2PA). To further enhance the protection against falsifications and deepfakes during biometric verification, future iterations of the system could integrate the Coalition for Content Provenance and Authenticity (C2PA) standard [8]. C2PA establishes a secure standard for digital provenance by adding cryptographically signed metadata to media files. This would act as a tamper-evident digital trail, proving when the media was captured, which device was used, and confirming that the content has not been altered. However, C2PA support in smartphones is currently not widespread, and recent studies have pointed out vulnerabilities [20].

9 Related Work

Decentralized Sybil-resistant solutions can be broadly categorized by their approaches to addressing Sybil resistance:

Social Graph. Employ social graph analysis algorithms [7, 10, 12, 18, 35, 40, 43, 48, 49] to detect Sybils. While effective, they often suffer from lack of anonymity and leaking data about which entities are connected. For example, **BrightID [6]** is a social identity network in which users form cryptographically signed connections that are stored on a proof-of-authority blockchain. While decentralizing connection data is essential for independent analysis, the social graph's public nature risks metadata leakage. Similarly, **Circles [24]** is a Universal Basic Income (UBI) protocol that uses transitive trust. Users issue and trust tokens, allowing transactions to route through a chain of mutually trusting peers. Sybil resistance is achieved by limiting transfer amounts based on trusted peers' balances, making tokens in poorly connected Sybil regions difficult to use. However, as with BrightID, the connections are public, raising the same metadata-leakage concerns.

Biometrics. Relies on unique physical characteristics and faces challenges from deepfakes, false acceptances, and data security. **World Network [45]** issues proof-of-human credentials by scanning users' irises with a proprietary hardware device called the **Orb** and storing identity commitments in a Merkle tree to enable ZKP-based verification. However, the Orb is a centrally manufactured device that creates an adoption bottleneck and introduces assumptions about centralized hardware trust.

Physical Presence. These approaches are based on the premise that a person cannot be in multiple places simultaneously. **Proof-of-Personhood [5]** binds physical entities to virtual identities through real-life, simultaneous, repeated pseudonym parties where attendees meet, exchange keys, and receive an identity token. However, this reliance on synchronized, globally coordinated physical parties severely limits scalability and usability. In contrast, De-SyRIS only requires a one-time, asynchronous in-person meeting between peers at their convenience. Alternatively, **Idena [22]** uses a proof-of-person consensus mechanism by conducting regular, online, simultaneous global Turing tests (FLIPS) that humans can pass but bots cannot. This synchronous testing suffers from rigid timing constraints (providing a poor user experience) and potential cheating.

Community Verification. Leverages verifiable claims that can represent individual reputation or mutual verification within Decentralized Autonomous Organizations (DAOs). **CanDID [26]** solves legacy compatibility by using Oracles, a decentralized committee, and Multi-Party Computation (MPC) to deduplicate and port centralized IDs (such as a Social Security Number) into decentralized verifiable credentials. While effective for compatibility, it fundamentally offshores Sybil resistance to legacy centralized institutions. **LinkDID [36]** is a web3-native identity scheme focusing on verifiable credentials, utilizing ZKPs to aggregate them and prevent cross-campaign linkage. Its Sybil resistance assumes that an entity accumulating many credentials from various trusted issuers is unlikely to be a Sybil. However, maintaining disjoint sets of identifiers with sufficient credentials is challenging but possible.

Mechanism Design. These systems utilize economic incentives or proof-of-resource protocols. Their main disadvantage is low inclusivity, as they often involve money and thereby benefit wealthy users. **Upala [32]** is an anti-Sybil system that measures identity reliability via a *Price-of-Forgery* score. Users form groups and assign dollar scores to members; users can *explode* their identity to claim the score in cash, permanently destroying it. Sybil resistance depends on the cost of forging an identity relative to the potential gain, meaning trust is based on economic criteria that inherently favor wealthy participants.

To address these limitations, De-SyRIS combines social graph analysis and facial biometrics. It preserves social connection and privacy, decentralizes facial verification, and maintains strict confidentiality of biometric data.

10 Conclusion

This paper presented De-SyRIS, a decentralized identity management system that effectively solves the Sybil resistance problem without relying on central authorities. By strategically synergizing hardware-backed facial biometrics with zero-knowledge social graph analysis, the system ensures global uniqueness while stringently preserving user privacy, anonymity, and unlinkability. The architectural design and proof-of-concept implementation demonstrate that computationally intensive operations, such as recursive

ZKPs for graph traversal, are practically feasible on modern hardware. Future research will focus on refining the decentralized reputation mechanisms that anchor the social graph and on exploring dynamic algorithmic thresholds for path verification to ease user bootstrapping.

References

[1] Md. Rayhan Ahmed, A. K. M. Muzahidul Islam, Swakkhar Shatabda, and Salekul Islam. 2022. Blockchain-Based Identity Management System and Self-Sovereign Identity Ecosystem: A Comprehensive Survey. *IEEE Access* 10 (25 10 2022), 113436–113481. doi:10.1109/ACCESS.2022.3216643

[2] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Cocco Weed, and Jason Yellick. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference* (Porto, Portugal) *(EuroSys '18)*. Association for Computing Machinery, New York, NY, USA, Article 30, 15 pages. doi:10.1145/3190508.3190538

[3] Marianna Belotti, Nikola Božić, Guy Pujolle, and Stefano Secci. 2019. A Vademecum on Blockchain Technologies: When, Which, and How. *IEEE Communications Surveys & Tutorials* 21, 4 (2019), 3796–3838. doi:10.1109/COMST.2019.2928178

[4] Aleksander Berentsen, Jeremias Lenzi, and Remo Nyffenegger. Fourth Quarter 2023. An Introduction to Zero-Knowledge Proofs in Blockchains and Economics. *Federal Reserve Bank of St. Louis Review* 105, 4 (Fourth Quarter 2023), 280–94. doi:10.20955/r.105.280-94

[5] Maria Borge, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, and Bryan Ford. 2017. Proof-of-Personhood: Redemocratizing Permissionless Cryptocurrencies. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 23–26. doi:10.1109/EuroSPW.2017.46

[6] BrightID. 2022. *BrightID*. whitepaper. Retrieved October 3, 2025 from https://www.brightid.org/whitepaper

[7] Qiang Cao and Xiaowei Yang. 2013. SybilFence: Improving Social-Graph-Based Sybil Defenses with User Negative Feedback. doi:10.48550/arXiv.1304.3819

[8] Coalition for Content Provenance and Authenticity. 2026. C2PA Specification. Version 2.4. Retrieved July 13, 2026 from https://spec.c2pa.org/specifications/specifications/2.4/index.html

[9] Victor Costan and Srinivas Devadas. 2016. Intel SGX Explained, Cryptology ePrint Archive, Report 2016/086. In *Intel SGX Explained, Cryptology ePrint Archive, Report 2016/086*. Retrieved July 13, 2026 from https://eprint.iacr.org/2016/086.pdf

[10] George Danezis and Prateek Mittal. 2009. Sybilinfer: Detecting sybil nodes using social networks.. In *Ndss*, Vol. 9. San Diego, CA, 1–15. Retrieved January 1, 2026 from https://crysp.uwaterloo.ca/courses/pet/W11/cache/netfiles.uiuc.edu/mittal2/www/sybilinfer-ndss09.pdf

[11] John R. Douceur. 2002. The Sybil Attack. In *Peer-to-Peer Systems*, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 251–260. doi:10.1007/3-540-45748-8_24

[12] Suhendry Effendy and Roland H. C. Yap. 2017. The Strong Link Graph for Enhancing Sybil Defenses. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 944–954. doi:10.1109/ICDCS.2017.117

[13] European Unified ID. 2026. European Unified ID: An open-source identity solution built for Europe. Retrieved August 10, 2026 from https://euid.eu/

[14] European Union. 2014. *Consolidated text: Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC*. Official Journal of the European Union, L 257, 28. 8. 2014, p. 73. Retrieved December 21, 2025 from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02014R0910-20241018

[15] European Union. 2024. *Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework*. Official Journal of the European Union, L 1183, 30.4.2024. Retrieved December 21, 2025 from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32024R1183

[16] Zhengze Feng, Ziyi Li, Hui Cui, and Monica T. Whitty. 2025. Identity Management Systems: A Comprehensive Review. *Information* 16, 9, 778 (2025), 778. doi:10.3390/info16090778

[17] Fhenix. 2026. Fhenix: Ethereum's privacy with Fully Homomorphic Encryption. Retrieved Accessed: 2026-09-25 from https://www.fhenix.io/

[18] Satoshi Furutani, Toshiki Shibahara, Mitsuaki Akiyama, and Masaki Aida. 2023. Interpreting Graph-Based Sybil Detection Methods as Low-Pass Filtering. *IEEE Transactions on Information Forensics and Security* 18 (18 1 2023), 1225–1236. doi:10.1109/TIFS.2023.3237364

[19] Aviral Goel and Yogachandran Rahulamathavan. 2025. A Comparative Survey of Centralised and Decentralised Identity Management Systems: Analysing Scalability, Security, and Feasibility. *Future Internet* 17, 1, 1 (2025), 1. doi:10.3390/fi17010001

[20] Enis Golaszewski, Neal Krawetz, Alan T. Sherman, Edward Zieglar, Sai K. Matukumalli, Roberto Yus, Carson L. Kegley, Michael Barthel, William Bowman, Bharg Barot, and Kaur Kullman. 2026. Verifying Provenance of Digital Media: Security Analysis of C2PA and its Implementation. Cryptology ePrint Archive, Paper 2026/804. https://eprint.iacr.org/2026/804

[21] Shengshan Hu, Minghui Li, Qian Wang, Sherman S. M. Chow, and Minxin Du. 2018. Outsourced Biometric Identification With Privacy. *IEEE Transactions on Information Forensics and Security* 13, 10 (10 2018), 2448–2463. doi:10.1109/TIFS.2018.2819128

[22] Idena. 2024. *Idena documentation*. Retrieved November 24, 2025 from https://docs.idena.io/

[23] Integritee. 2024. *Integritee Network Docs*. Technical Report. Integritee, Inc. Retrieved July 22, 2026 from https://github.com/integritee-network/docs/tree/2f7e468b1a85534ff2da115706f850cf846ae60d

[24] Martin Köppelmann, Paul Boes, and Friederike Ernst. 2024. *Circles – money for a multipolar world*. whitepaper. Retrieved October 3, 2025 from https://whitepaper.aboutcircles.com/

[25] Ryan Lavin, Xuekai Liu, Hardhik Mohanty, Logan Norman, Giovanni Zaarour, and Bhaskar Krishnamachari. 2024. A Survey on the Applications of Zero-Knowledge Proofs. arXiv:2408.00243 [cs.CR] doi:10.48550/arXiv.2408.00243

[26] Deepak Maram, Harjasleen Malvai, Fan Zhang, Nerla Jean-Louis, Alexander Frolov, Tyler Kell, Tyrone Lobban, Christine Moy, Ari Juels, and Andrew Miller. 2021. CanDID: Can-Do Decentralized Identity with Legacy Compatibility, Sybil-Resistance, and Accountability. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1348–1366. doi:10.1109/SP40001.2021.00038

[27] Alexander Mühle, Andreas Grüner, Tatiana Gayvoronskaya, and Christoph Meinel. 2018. A survey on essential components of a self-sovereign identity. *Computer Science Review* 30 (11 2018), 80–86. doi:10.1016/j.cosrev.2018.10.002

[28] Oasis Protocol Foundation. 2020. *The Oasis Blockchain Platform*. Technical Report. Oasis Labs. White paper. Retrieved April 16, 2026 from https://assets.website-files.com/5f59478e350b91447863f593/628ba74a9aee37587419cf65_20200623%20The%20Oasis%20Blockchain%20Platform.pdf

[29] Oasis Protocol Foundation. 2026. Oasis Network Documentation. Retrieved April 16, 2026 from https://docs.oasis.io

[30] OpenID Foundation. 2026. How OpenID Connect Works. Retrieved August 10, 2026 from https://openid.net/developers/how-connect-works/

[31] Polygon Zero Team. 2022. Plonky2: Fast Recursive Arguments with PLONK and FRI. (7 9 2022), 16. Retrieved May 11, 2026 from https://docs.rs/crate/plonky2/latest/source/plonky2.pdf

[32] Peter Porobov. 2019. *Upala*. Retrieved December 2, 2025 from https://docs.upala.id/en/latest/

[33] Arun Sekar Rajasekaran, Maria Azees, and Fadi Al-Turjman. 2022. A comprehensive survey on blockchain technology. *Sustainable Energy Technologies and Assessments* 52 (2022), 102039. doi:10.1016/j.seta.2022.102039

[34] Alex Seto, Oytun Kuday Duran, Samy Amer, Jalen Chuang, Stephan van Schaik, Daniel Genkin, and Christina Garman. 2025. Wiretap: Breaking server sgx via dram bus interposition. In *2025 SIGSAC Conference on Computer and Communications Security (CCS '25)*. Association for Computing Machinery, 708–722. doi:10.1145/3719027.3765204

[35] Lu Shi, Shucheng Yu, Wenjing Lou, and Y. Thomas Hou. 2013. SybilShield: An agent-aided social network-based Sybil defense among multiple communities. In *2013 Proceedings IEEE INFOCOM*. IEEE, 1034–1042. doi:10.1109/INFCOM.2013.6566893

[36] Rui Song. 2024. LinkDID: A Privacy-Preserving, Sybil-Resistant and Key-Recoverable Decentralized Identity Scheme. arXiv:2307.14679 doi:10.48550/arXiv.2307.14679

[37] Manu Sporny, Dave Longley, David Chadwick, and Ivan Herman. 2025. *Verifiable Credentials Data Model v2.0*. Technical Report. World Wide Web Consortium. Retrieved January 16, 2026 from https://www.w3.org/TR/vc-data-model/

[38] Manu Sporny, Dave Longley, Markus Sabadello, Drummond Reed, Orie Steele, and Christopher Allen. 2022. *Decentralized Identifiers (DIDs) v1.0*. Technical Report. World Wide Web Consortium. Retrieved January 16, 2026 from https://www.w3.org/TR/did-1.0/

[39] Nick Szabo. 1996. Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought,(16)* 18, 2 (1996), 28.

[40] Nguyen Tran, Jinyang Li, Lakshminarayanan Subramanian, and Sherman S.M. Chow. 2011. Optimal Sybil-resilient node admission control. In *2011 Proceedings IEEE INFOCOM*. IEEE, 3218–3226. doi:10.1109/INFCOM.2011.5935171

[41] Unique Identification Authority of India. 2025. *Aadhaar*. Retrieved December 22, 2025 from https://uidai.gov.in/en

[42] Sa Wang. 2025. *What Is Entity Resolution: Techniques, Tools & Use Cases*. Retrieved December 21, 2025 from https://www.puppygraph.com/blog/entity-resolution

[43] Wei Wei, Fengyuan Xu, Chiu C. Tan, and Qun Li. 2013. SybilDefender: A Defense Mechanism for Sybil Attacks in Large Social Networks. *IEEE Transactions on*

Parallel and Distributed Systems 24, 12 (11 1 2013), 2492–2502. doi:10.1109/TPDS.2013.9

[44] Carter Woetzel. 2016. Secret Network: A privacy-preserving secret contract & decentralized application platform. *Secret Foundation* (2016), 16. Retrieved July 22, 2026 from https://resources.cryptocompare.com/asset-management/401/1696324751265.pdf

[45] World. 2026. *World whitepaper.* Technical Report. Retrieved April 3, 2026 from https://whitepaper.world.org/

[46] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. 2019. Blockchain Technology Overview. *CoRR* abs/1906.11078 (2019). arXiv:1906.11078 doi:10.6028/NIST.IR.8202

[47] Hang Yin, Shunfan Zhou, and Jun Jiang. 2022. Phala network: A secure decentralized cloud computing network based on Polkadot. *Phala Network, Tech. Rep* (7 3 2022), 15. Retrieved July 22, 2026 from https://resources.cryptocompare.com/asset-management/259/1665667815920.pdf

[48] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. 2008. SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 3–17. doi:10.1109/SP.2008.13

[49] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham D. Flaxman. 2008. SybilGuard: Defending Against Sybil Attacks via Social Networks. *IEEE/ACM Transactions on Networking* 16, 3 (30 6 2008), 576–589. doi:10.1109/TNET.2008.923723

[50] Guy Zyskind, Yonatan Erez, Tom Langer, Itzik Grossman, and Lior Bondarevsky. 2025. FHE-Rollups: Scaling Confidential Smart Contracts on Ethereum and Beyond. In *Proceedings of the 6th ACM International Symposium on Blockchain and Secure Critical Infrastructure* (Singapore, Singapore) *(BSCI '24)*. Association for Computing Machinery, New York, NY, USA, 1––9. doi:10.1145/3659463.3660031