De-SyRIS: A Decentralized Sybil-Resistant Identity System without a Trusted Credential Issuer

Michal Ľaš
Brno University of Technology
Brno, Czech Republic
xlasmi00@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 an identity management architecture that ensures robust Sybil resistance and is fully decentralized, unlike the systems currently in widespread use. The proposed solution synergizes facial biometric verification with anti-Sybil social graph analysis. By leveraging blockchain technology, zero-knowledge proofs, and trusted execution environments, the system ensures the uniqueness, anonymity, and privacy of registered users without relying on centralized authorities. The operation of the proposed system was demonstrated through simulations of a network of 100 entities across three social graph models and through a proof-of-concept implementation that utilizes fast Plonky2 zero-knowledge proofs. To our knowledge, this is the first work that proposes a Sybil-resistant identity system without sacrificing decentralization and privacy, while not relying 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 [27], EUID [13]), or decentralized identities [35]. Typically, these identities are managed by centralized authorities, which introduces significant privacy, security, and censorship risks [16, 18]. While self-sovereign identity (SSI) approaches have emerged to grant users complete control over their identity-related data, they often rely on centralized credential issuance [1, 24]. To prevent Sybil attacks [11] – where malicious actors forge numerous identities to subvert a network – most systems verify identity uniqueness using government-issued documents or phone numbers [14, 15, 38]. Consequently, because uniqueness relies on a trusted third party, the core trust mechanism remains centralized even within seemingly decentralized ecosystems. This article explores the 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 strong defense against sophisticated biometric spoofing and isolated Sybil clusters.
(2) We utilize a decentralized privacy-preserving smart contract platform (equipped with TEE) 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, 30, 43] 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 [36], which are self-executing programs that enforce rules autonomously, eliminating the need for intermediaries.

Identity Management Systems. Identity Management Systems (IDMSs) [16, 18] 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 [35] (DIDs) and Verifiable Credentials [34] (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, 22] is a cryptographic protocol involving two parties: a *prover* and a *verifier*. The protocol enables the prover to convince the verifier that a specific computational statement is true without revealing any information beyond the statement's validity. 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 enables 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, 25, 41, 44]. 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, 27, 38] 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 [39] 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 [42] 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 seek to propose a pure decentralized Sybil-resistant identity system without reliance on trusted credential issuers.

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 random walk 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: a **fast-mixing honest region** and a **densely connected Sybil region**, separated by a sparse set of attack edges. The honest region must be defined by known, trusted nodes to distinguish it from the Sybil region. Structure of the regions alone is not sufficient, since the attacker can connect its controlled Sybil nodes in any way and effectively mimic the structure of the honest regions.

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

(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.

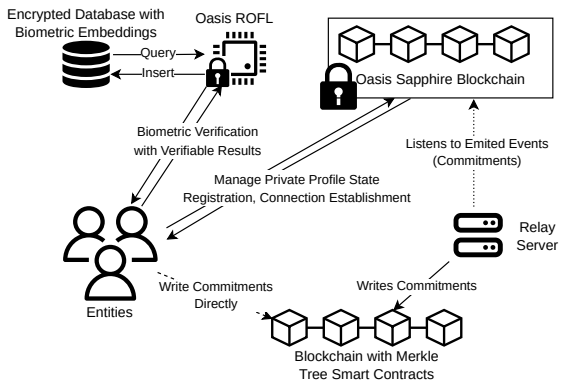

Figure 1: Actors of the proposed Sybil-resistant system and their interactions.

(2) **Connection Establishment:** Registered entities form mutual, private connections by physically meeting and verifying each other's biometric uniqueness with the utilization of the stored template from step (1) and provided identity information. These peer-to-peer endorsements construct the edges of the decentralized social graph.
(3) **Proof of Connection with Honest Entities (Path Proving):** To demonstrate integration into the honest region of the network and prove Sybil resistance, entities generate recursive ZKPs of their connections. This process, called *private random walks*, allows entities to construct continuous paths through the graph without ever revealing the identity of connected entities and the graph's topology to the public.

To facilitate these processes while guaranteeing privacy, the system employs the Oasis Network [25, 26], which is a 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 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 the 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 extremely difficult to accomplish. Conversely, the biometric requirement raises the cost and complexity of identity generation, preventing the effortless mass-creation of fake nodes that a purely graph-based system might suffer from.

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: • ID_A: The unique identifier of the entity A (derived from public key PK_A). • SK_A: The private key of entity A. • FS_A: The facial scan of entity A. • M_A: The C2PA manifest for the FS_A or P_A. • E_A: The multidimensional biometric embedding extracted from FS_A or P_A. It can have the upper index s *stored*, or c *captured*. • $H(E_A)$: The hash of the normalized embedding E_A. • t_{M_A}: A timestamp parameter that is part of the M_A. • CC_{AB}: The connection commitment between A and B. • S_{AB}^{CC}: A shared secret salt used to blind the connection commitment. • CC_{AB}^{s}: A salted connection commitment between A and B. • $Sig_A\{content\}$: 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.

Coalition for Content Provenance and Authenticity (C2PA). Since users can scan their facial biometric data on their own devices, it is essential to protect against forgeries, tampering, and deepfakes. C2PA [8] has established a secure standard for digital provenance by adding cryptographically signed metadata (e.g., date, time, GPS coordinates) to media files such as photos and videos. This C2PA manifest acts as a tamper-evident digital trail, proving when the media was captured, which device and application were used, and confirming that the content has not been altered. Within the proposed architecture, C2PA manifests are required during the biometric registration and connection establishment phases to ensure all submitted facial scans represent genuine, real-time captures.

However, C2PA certificates are relatively new, and widespread device support is currently lacking. To maintain compatibility with unsupported hardware, the Oasis ROFL execution layer must temporarily perform alternative, robust biometric verification.

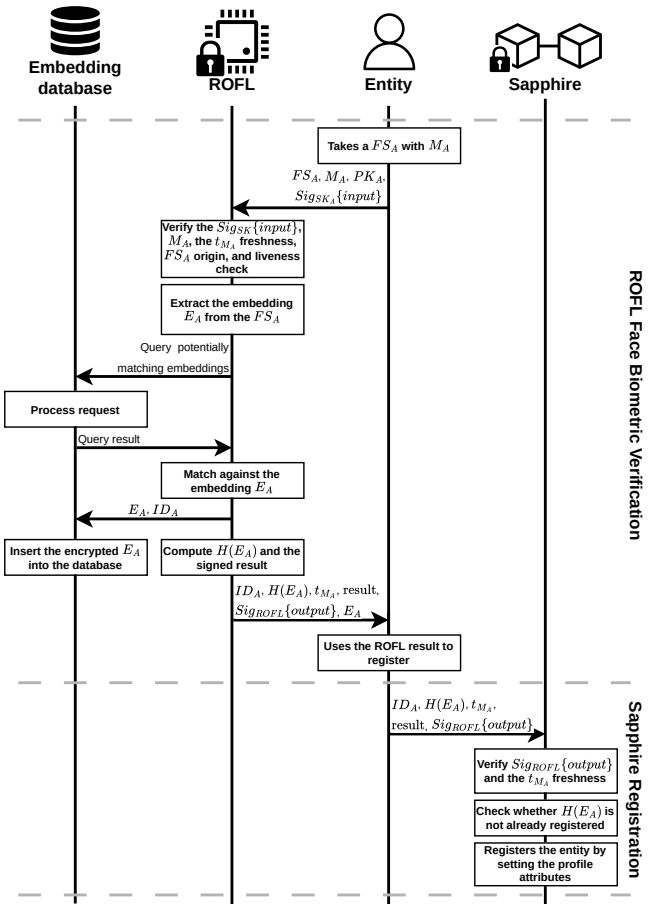

Figure 2: De-SyRIS profile registration process.

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 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 facial scan (FS_A), signs it with a C2PA manifest (M_A), and sends it to ROFL for verification. The process can be summarized as follows:

- **Input:** FS_A, M_A, PK_A, and $Sig_{SK_A}\{input\}$.
- **Output:** ID_A, $H(E_A)$, t_{M_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) Verify M_A to ensure the integrity, the freshness of the t_{M_A} (for example, no older than one hour), and that the FS_A was captured by a verified application.
(3) Perform liveness checks on FS_A. This can involve complex liveness verification, such as interactive challenges issued by the ROFL enclave to the entity.
(4) Extract the biometric embedding E_A from the FS_A.
(5) Query the encrypted database for potentially matching embeddings.
(6) Decrypt the queried embeddings within the TEE and match them against the embedding E_A currently being registered.
(7) If no match is found, insert the encrypted E_A into the database.
(8) 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 identical result.
(9) Return a signed result $Sig_{ROFL}\{output\}$ that includes ID_A, $H(E_A)$, t_{M_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_{M_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_{M_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 already existing record).

In the case of limited registration, the entity must obtain community endorsements from previously registered users to achieve full status. Endorsements serve as a secondary, community-driven layer of Sybil resistance designed to overcome the natural false-positive rate of facial recognition. To prevent indiscriminate vouching and preserve the integrity of the social graph, each fully registered user is assigned only a single endorsement to give. A user with a limited profile must reach a specific endorsement threshold before their biometric embedding is inserted into the database and their 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 photographs of each other (A takes P_B, B takes P_A) using an authorized mobile application. These photographs are also signed with

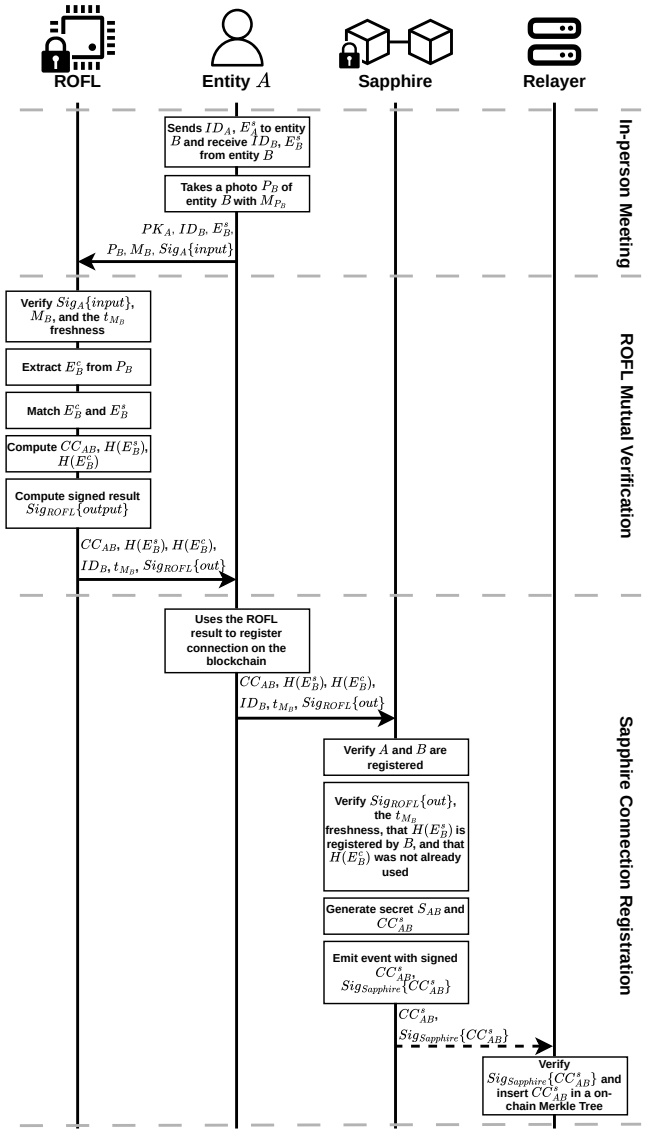

Figure 3: Connection establishment process between two entities from the perspective of Entity A.

the C2PA manifest (M_A, M_B). 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.

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, P_B, M_B, E_B^s, and $Sig_A\{input\}$.
- **Output:** CC_{AB}, $H(E_B^s)$, $H(E_B^c)$, ID_B, t_{M_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) Verify M_B to ensure the integrity, the freshness of the t_{M_B} (for example, no older than 10 minutes), and that it was captured via an authorized application.
(3) Extract embedding E_B^c from P_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 $CC_{AB} = H(\min(ID_A, ID_B) \parallel \max(ID_A, ID_B))$[1].
(6) Return a signed result $Sig_{ROFL}\{output\}$ including CC_{AB}, $H(E_B^s)$, $H(E_B^c)$, ID_B, and t_{M_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 both A and B are registered, the validity of signature $Sig_{ROFL}\{output\}$, the freshness of the t_{M_B}, that $H(E_B^s)$ is 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) Generates a random secret S_{AB}^{CC} serving as a shared salt and the salted commitment $CC_{AB}^s = H(CC_{AB} \parallel S_{AB})$.
(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.

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 Random Walks** and ZK **Proof of Paths**.

Reputation. To distinguish the honest region from the Sybil region, the system relies on external, decentralized reputation scores. Notably, only a small fraction of entities require this external reputation, as the rest can be verified through transitive connections.

To incentivize participation, an entity's total reputation is divided into **external** reputation and **intrinsic** reputation (earned via protocol activity). Both components have maximum limits, and their sum is committed to an on-chain Merkle tree to enable verification in ZKPs.

5.3.1 ***ZK Random Walks.*** Entities construct **paths** through the social graph by proving their connection to the previous entity and appending their own identity and reputation score to the path. They then forward this proof to the next connection.

This process is illustrated in Figure 4. The diagram above shows how the entities are connected. The edges indicate the path that

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

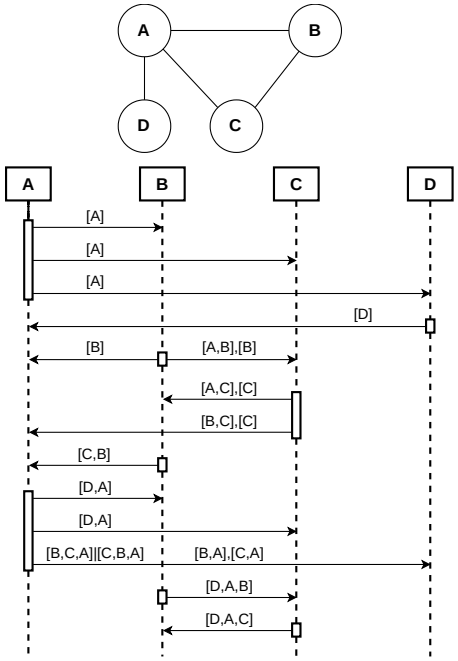

Figure 4: Step-by-step construction of a ZK random walk among four entities.

an entity follows. For example, $[A, B, C]$ denotes a path 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. All entities repeat this process, regardless of the order.

Valid paths require a length of $O(\log n)$ 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 has fewer than $2 \times k$ 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.

However, false negatives may occur if the entity performing the ZK Proof of Paths has a collision in the Bloom filter between its set bits and the bits set by an entity that appears in multiple paths. To minimize bit collisions, the filter must be sparse. For example, for paths of length 10, a 4096-bit filter with 3 hash functions yields error rates around $\approx$ 0.05% (false positives) and $\approx$ 0.01% (false negatives). Ultimately, this verifies path distinctness, preventing Sybil attackers from exploiting single bottlenecks while maintaining absolute topology privacy.

5.3.2 ***ZK Proof of Paths.*** Once an entity has gathered the required number of distinct paths (sets of entities on all paths have to be disjoint), it can create a ZKP that aggregates these proofs into a single one. 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. However, tracking these changes poses challenges for anonymity and privacy. To address this, De-SyRIS 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. An epoch can last between 3 and 7 days, depending on the system's intended strictness.

5.5 Properties of Proposed System

The proposed system addresses the key desired properties of 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 Security Analysis

The most significant attacks occur when an attacker acquires a large number of Sybil identities, either by attacking the social graph analysis mechanism or by stealing other users' identities. At the same time, it is important to protect privacy and anonymity and prevent tracking of users' actions, as this could lead to censorship.

6.1 Facial Biometric Fraud and Forgeries

The threat of Sybil creation via sophisticated physical masks, AI-generated deepfakes, or maliciously altered photographs is primarily addressed by the C2PA standard. The protocol mandates that registration photos contain cryptographically signed metadata generated securely by the smartphone's trusted hardware. Consequently, it requires significant hardware-level exploitation to bypass this defense.

Since support for the C2PA standard in smartphones is not yet very widespread, it can theoretically be omitted. Registration is the most critical process in this case. Verification of registered biometric data occurs in a decentralized manner by other entities when establishing a connection. However, an attacker can create fake

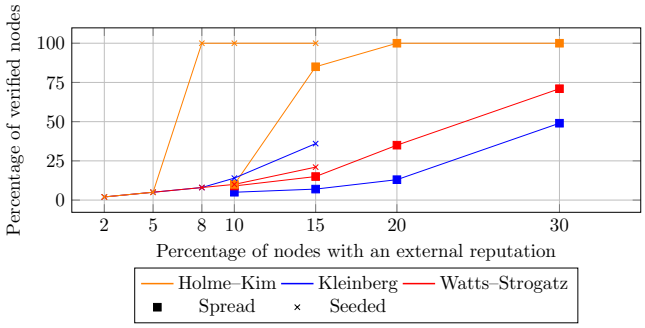

Figure 5: Results of the entity verification simulation based on the number of entities with external reputation.

identities and link them together, but will have difficulty establishing connections with other honest entities. Failing to use C2PA certificates and robust liveness detection during registration thus reduces the defense against Sybil entities to merely analyzing the social graph.

6.2 Attacks on Social Graph Analysis

For a Sybil cluster to be recognized by the system, the attacker must acquire k distinct attack edges (connections) to the honest region. However, a coordinated group of malicious actors could leverage their real, highly reputable identities as gateways to legitimize a massive Sybil cluster.

Additionally, the proposed system currently lacks a mechanism to prevent or penalize entities that intentionally establish connections with any entities—including unknown ones that may be Sybil accounts—thereby enabling them to obtain verification. It is essential to create a mechanism that would incentivize entities to establish connections only with entities they trust.

6.3 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 random walk 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.

7 Evaluation

The evaluation of the proposed De-SyRIS focuses on assessing its architectural properties against the foundational requirements of Sybil-resistant identity systems. Additionally, a Proof of Concept (PoC) for the system was developed, along with a simulation to track the distribution of reputation scores and the ability of entities to receive verification within the social graph.

7.1 Simulation of the Social Network Analysis

The simulation aimed to model trust distribution among entities without an external reputation using a graph of 100 entities over 30 epochs. Three graph models – Holmes–Kim, Watts–Strogatz, and Kleinberg small-world – were utilized to reflect real-world network properties. The results are presented in the graph in Figure 5.

Two trust distribution modes were tested: spread and seeded. In spread mode, a percentage of entities receive reputation from a normal distribution, capped at the maximum external reputation. In seeded mode, fewer entities were assigned the maximum reputation. These two modes represent how a reputation system can function. Spread refers to a system in which all entities can attain a specific reputation rating. Seeded refers to a system that includes highly vetted entities known to be trustworthy.

The ability of entities to acquire a reputation depends on connectivity. Spread mode needs 15-30% of entities with external reputation, while seeded mode requires 8-10% in the Holme–Kim model. Other models are unsuitable due to weak connectivity.

Although attacks can be simulated, their success depends on factors such as the number and length of required paths and the attacker's connections. A more detailed analysis is needed to evaluate the algorithm's effectiveness in preventing Sybil entities from obtaining verification.

7.2 ZK Proof Performance

De-SyRIS contains two non-trivial ZKPs, and – especially in the case of ZK Random Walks – the speed of these proofs is important because they are generated frequently. The Plonky 2 [28] scheme was selected for these ZKPs for its highly efficient recursion and optimized Poseidon hashing. Two circuit designs for the random walks 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 random walks. The results are shown in the Table 1. Note that the ZK Random Walks in the PoC implementation do not use the proposed approach with Bloom filters; instead, they use a simple array of nullifiers.

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.

Example	Circuit		Proof		
	Comp.	Size	Prove	Ver.	Size
ZK Random Walk Linear	1.56s	8KB + $l \times$ 4.2MB	328.56ms	4.44ms	124KB
ZK Random Walk 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 Random Walks 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.

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.

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 [20] was discontinued following the Intel SGX WireTap vulnerability [31], which Intel will not patch as it assumes physical security. Consequently, the secure functioning of TEEs in decentralized environments remains an open question.

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, 17, 32, 37, 40, 45, 46] to detect Sybils. While effective, they often suffer from slow bootstrapping and a lack of anonymity. For example, **BrightID [6]** is a social identity network where users form cryptographically signed connections stored on a proof-of-authority blockchain. While decentralizing connection data is essential for independent analysis, the public nature of the social graph risks metadata leakage. Similarly, **Circles [21]** 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 [42]** 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 pseudonym parties where attendees meet, exchange keys, and receive an identity token. However, this reliance on synchronized physical parties severely limits global scalability and usability. Alternatively, **Idena [19]** 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 [23]** 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 [33]** 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 [29]** 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] 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

[18] 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

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

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

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

[22] 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

[23] 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

[24] 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

[25] 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

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

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

[28] 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

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

[30] 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

[31] 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

[32] 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

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

[34] 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/

[35] 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/

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

[37] 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

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

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

[40] 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

[41] 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

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

[43] 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

[44] 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

[45] 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

[46] 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