RARIMO: A PRIVACY-FIRST (ZK) SOCIAL PROTOCOL

WHITEPAPER

May 28, 2024

ABSTRACT

Rarimo is a privacy-first (zk) social protocol that enables the development of a new generation of social apps. Solutions built on Rarimo allow users to operate incognito while preserving their history of actions, connections, and identity attributes. Users can selectively disclose statements while hiding other aspects of their social graph.

1 The publicity vs anonymity dilemma

Public social networks have quickly become the primary way we interact online. They excel at building communities and fostering engagement but at the cost of exposing your identity and activities online. This exposure is increasingly concerning amid rising censorship, surveillance, and impersonation scams.

On the other hand, anonymous chatrooms and imageboards sharply contrast these publicity-focused platforms. They allow for greater freedom of expression by keeping identities private. However, this anonymity can impede community growth and meaningful collaboration.

Let's explore the current landscape of social protocols and envision the next generation of social apps that offer the best of both worlds.

1.1 Public networks

Public networks often mirror the dynamics of real-world communities, featuring hierarchical structures where influence is tied to members' reputations. This reputation may be implicit, as evidenced by a member's recognition within the community, or explicit, through mechanisms such as Reddit karma or Discord badges. Entry into these communities is generally gated, requiring consensus among current members or some form of social proof, thereby limiting accessibility to outsiders. Such measures bolster social cohesion and foster deep, sustained cooperative interactions among members. Nevertheless, this framework can induce rigidity and promote **self-censorship** as individuals strive to preserve their status and adhere to established community norms.

Traditional public networks are centralized, and governed by administrators who impose stringent policies on permissible behaviors. These platforms retain complete control over the social graphs and monetize this data through transactions with advertisers and governmental entities, leading to potential censorship concerns.

Conversely, Web3 public networks advocate for user empowerment by granting individuals direct control over their social graphs. While these networks forgo privacy, they offer a decentralized record of social interactions, historical data, and a reputation system. However, the transparency of these networks raises significant security issues. The public accessibility of social graphs allows for analysis and replication by AI tools, potentially exposing social interactions to sophisticated bot infiltrations [@St24, DFMMR16].

1.2 The wild west of anonymity

Chaos reigns in the realm of anonymous platforms [BMHH+21]. This world operates without needing identity disclosure or a vetting process, eliminating traditional gatekeeping and cultivating an egalitarian environment where reputation considerations do not constrain actions. This level of freedom of expression pushes the limits of diverse and often unconventional ideas, with notable instances found in imageboards like 4chan and IRC chatrooms infamous for their anarchic nature.

However, this model has its drawbacks. The lack of mechanisms to verify statements, prove ownership, or establish enduring relationships poses significant challenges. The ephemeral nature of identities in these settings restricts the ability to coordinate actions or manage groups anonymously. Without the foundation of a reputation or historical record, it becomes impractical to enforce deterministic rules or filter interactions based on identity-related criteria, undermining the potential for building trust and accountability within the community.

1.3 Designing the next generation of social protocols

Current social protocols sacrifice anonymity for publicity or reputation building for privacy. The recent advancements in technology allow us to combine the strengths of both paradigms. Future applications should enable users to engage freely in activities, establish relationships, and assert ownership within a secure framework that respects their privacy. Here, we outline the desired characteristics for a next-generation social protocol:

Self-Sovereignty: Users must have full control over their accounts, eliminating dependency on third parties.

Decentralization: The protocol should operate independently of centralized services, reducing the risks of censorship and surveillance.

Provability: Users should be able to prove statements about their social graph, maintaining anonymity.

Private Social Graph: It should be technologically infeasible to map or analyze a user's relations or actions using open-source intelligence (OSINT) techniques or artificial intelligence (AI) tools.

Rules-Based Group Membership: A rules-based algorithm should govern group membership management, avoiding the arbitrary influence of moderators.

Table 1: Comparison of approaches for building social protocols

Criteria	Public networks	Anonymous platforms	Rarimo social apps
Account management	Centralized(except Web3)	Centralized or absent	Self-sovereign
Decentralization	Mostly centralized(except Web3)	Mostly centralized	Decentralized infra and governance
Provability	Based on public social graph	Limited	Based on ZK proofs
Social graph	Public	Absent	Private (part. knowledge)
Group management	Centralized	Absent	Rules-based and deterministic

2 Rarimo Protocol

Rarimo is a distinct social protocol that focuses on privacy and introduces a new framework for building a new generation of social applications. Users can freely interact and expand their social interactions while remaining incognito.

With Rarimo, all identity credentials, relationships, and activity histories are kept on the user's side, removing the need for third-party involvement. Social apps interact with the private social graphs using zero-knowledge proofs. This approach lets users selectively disclose parts of their social graph while keeping the rest of the data secure.

This design supports anonymous interactions and helps users maintain their connections and reputations. Additionally, it allows them to set programmable rules for community management without admins or reliance on central arbiters. Rarimo offers a flexible and secure framework for those seeking privacy in their social interactions.

The following sections will detail the foundational elements of the Rarimo protocol.

2.1 Commitments, the backbone of private interactions

In the Rarimo protocol, a cryptographic commitment is a crucial low-level primitive that binds specific data while concealing its content. This function is vital in standardizing all statements at the protocol level. Such standardization is essential for maintaining the uniformity of the structure and update mechanisms of social trees, a concept introduced later in this paper. The uniform structure of social trees facilitates the creation of efficient inclusion proofs, which is particularly beneficial when these proofs need to be aggregated, for example, when a user must demonstrate their inclusion within a set of trees. This method ensures both the integrity and privacy of user data within the network.

The construction of a commitment is as follows:

$$Comm = hash(statement||salt) \tag{1}$$

The $statement$ can represent any data, including some algorithms and programs. The $salt$ is an additional value generated by the commitment initiator that allows the statement to be blinded. Usually, the $salt$ is random, but in some cases, it can be deterministic (i.e., for achieving uniqueness).

When constructing a commitment, the user creates an irreversible anchor for the statement while keeping its content private. The following actions should be performed to selectively disclose information about the underlying statement:

- Prove that the commitment is part of a particular tree. Some trees can be built off-chain with time stamping only their root values.
- Prove the knowledge of the statement and the salt (two values that were used for commitment construction)
- Prove that the statement satisfies particular criteria. The complexity of the criteria can range from a statement revealing to provable queries without disclosure.

2.2 Rarimo Core, the state persistence and cross-chain communication layer

Rarimo Core functions as a blockchain layer dedicated to collecting and timestamping identity-related events and organizing the social forest. This architecture is meant for constructing and efficiently propagating the confidential social graph (detailed in Section 3) across connected networks. A group of validators support the integrity and reliability of Rarimo Core. These validators are selected through a Delegated Proof of Stake (DPoS) mechanism.

While social applications can operate on any blockchain, Rarimo Core facilitates the integration of a unified social layer across different environments. Once consensus is achieved among the validators, the finalized state of the graph is secured using a Threshold Signature Scheme (TSS). This cryptographic method allows the state to be authenticated and subsequently propagated to other chains without requiring modifications at the protocol level, thereby ensuring both scalability and security in cross-chain communications.

2.3 Social Forrest, the efficient data structure for commitments

Storing raw commitments directly on the blockchain poses significant challenges in terms of efficiency, scalability, and privacy. Specifically, revealing a transaction sender when a user proves data from a specific commitment can compromise privacy. Although it is technically feasible to create proof that a commitment is stored on-chain, this approach remains complex and inefficient. These issues can be mitigated by incorporating the commitments into specialized trees, each designed with distinct purposes and characteristics.

Rarimo Protocol defines the following tree types 1:

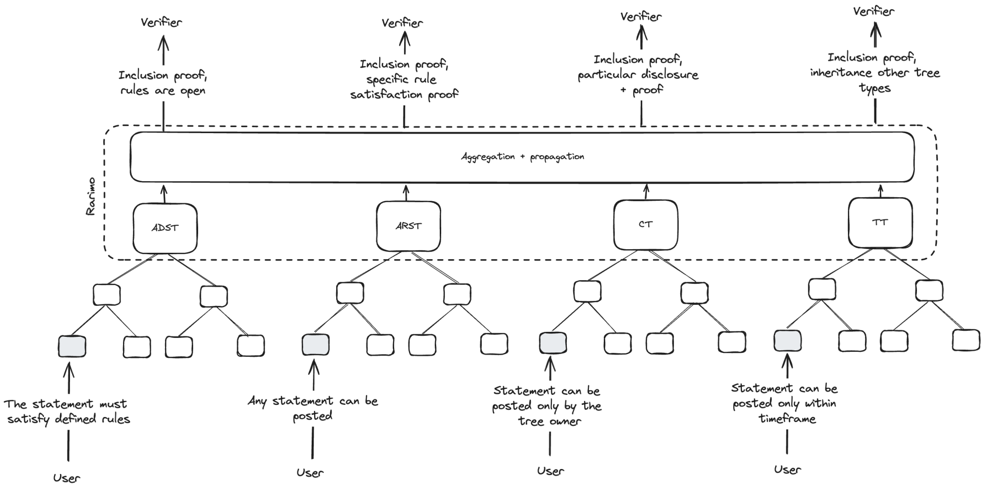
Verifier
Inclusion proof, rules are open
Verifier
Inclusion proof, specific rule satisfaction proof
Verifier
Inclusion proof, particular disclosure + proof
Verifier
Inclusion proof, inheritance other tree types
Aggregation + propagation
Rarimo
ADST
ARST
CT
TT
The statement must satisfy defined rules
User
Any statement can be posted
User
Statement can be posted only by the tree owner
User
Statement can be posted only within timeframe
User

Figure 1: Tree types.

1. **Statement Trees (ST)**. These trees accommodate basic statements without hierarchical relations and time bounds. They are further subdivided into:
   (a) **Adjustable Statement Trees (ADST)**. This sub-type allows setting rules for adding commitments to the tree. To add the commitment into the **ADST** tree, the user must prove the statement satisfies rules (partial knowledge). In this subtype, predefined rules govern the addition of commitments. Users must prove that their statements satisfy these rules(partial knowledge).
   (b) **Arbitrary Statement Trees (ARST)**. This subtype imposes no validation rules on the data added. It acts as an open outlet where any commitment can be submitted and later revealed alongside its validation rules. Like a permissionless wall where anyone can write what they want.
2. **Credential Trees (CT)**. These trees establish hierarchical relationships (one-to-many) between the issuer and the owners of credential commitments. Each tree has an owner, although multiple signatures (multisig) are supported.
3. **Time Trees (TT)**. Dedicated to a specific time range, these trees restrict the addition of new commitments after a predetermined point. They can be based on either ST or CT structures.

By segregating commitments into different tree types based on their properties, the protocol allows users to select the level of rule enforcement required for their use case. This segregation provides structured yet flexible interaction options, making the protocol adaptable to various applications and dynamic requirements.

Furthermore, the flexible structure of the social forest is designed to accommodate future security enhancements, such as new cryptographic signatures and proving schemes. As technology and security needs evolve, the protocol can adapt seamlessly without necessitating major overhauls, much like how real trees adapt to their environment. This design facilitates a balance between efficiency and advanced security.

Here are several examples of how different applications can be built on top of the social forest:

- Users seeking to create a permissionless chat may utilize the **ARST** tree. Each message is encapsulated within a corresponding commitment. For anonymity, participants might opt for random salts and signatures, discarding them after usage. Deterministic salt and consistent keypairs for signatures can be used to prove the message sequence later.
- To establish a chat accessible only to users meeting specific criteria, the **ADST** tree should be employed. Users must provide proof of eligibility when posting messages,

with the eligibility criteria set transparently by the chat's creator and enforced via smart contract.

- Entities desiring to act as identity providers, such as authoritative organizations issuing verifiable credentials, should use the **CT** tree. This tree type is exclusively manageable by its owner, unlike the **ARST** and **ADST** trees, which their initial creators manage.
- For events confined to a specific timeframe, such as petition signings or periodic check-ins for liveness proofs, the **TT** tree is most suitable.

3 Incognito social graph

Several leaves of the social forest can belong to the same identity, with the identity owner able to prove their connection. These leaves and their relations can serve as criteria for forming social groups.

This collection of social forest leaves linked to a single identity may be defined as a private social metagraph. The user has the option to selectively reveal parts or properties of this metagraph to meet the criteria for group membership.

Examples of such conditions include:

- Proving the possession of a credential and attestation at a specific time (or continuously).
- Demonstrating that some data within the commitment passed verification defined by a particular DApp or verifier.
- Giving or receiving an attestation to or from another identity owner (of the whole identity or a specific attribute).
- Proving that a particular user initiated specific actions.
- Proving that a specific group initiated certain actions and that a predefined threshold of group members meets specific requirements.
- Proving inclusion or exclusion in specific lists or groups.
- Proving inclusion or exclusion in specific lists or groups.
- Proving that sent or received attestations originate from an identity or service that meets certain criteria without revealing the owner's identity.

These interactions are depicted in Figure **??**. External auditors are limited to viewing only the trees, while the relations between the commitments remain known exclusively to the identity owner. The owner can selectively disclose parts of the graph, which is essential for group eligibility verification.

4 The road ahead

Implementing the social metagraph within the Rarimo protocol presents several critical challenges that must be solved to ensure proper functioning and an optimal user experience of social apps. These challenges include:

- **Standardization**: Although the Rarimo protocol offers flexibility in how statements and commitments are managed, there is a need for standardization and unification across typical features of social applications. Developing a comprehensive toolkit to facilitate the basic mechanics of these applications is an essential next step.
- **Advanced and Efficient Security**: Rarimo currently supports the generation of SNARK proofs. However, incorporating more robust proving schemes and supporting more complex queries, especially recursive ones, would significantly improve the security and flexibility of the system. As most user interactions and proof verifications occur on personal devices, optimizing the efficiency of the ZK computations is also essential.
- **Recovery Difficulties**: Users must manage substantial amounts of data linked to their metagraph, complicating account recovery and synchronization of profile data across

different devices. This issue necessitates reliable data backup and secure transfer mechanisms for safe and user-friendly account recovery.

- **Calculations over the Metagraph**: Designing a mechanism that can perform computations or extract insights from the metagraph without compromising specific details or sensitive information poses a significant challenge. Such a mechanism must handle encrypted and anonymized data.

Addressing these challenges is crucial for the successful implementation and widespread adoption of the Rarimo protocol.

5 Conclusion

Rarimo protocol introduces a significant shift in how social networks operate by making the social graphs private. Using zero-knowledge proofs and decentralized infrastructure, Rarimo allows users to stay incognito but still prove their actions and relationships, addressing the main issues in both public and anonymous platforms. This protocol emphasizes giving users complete control over the social graph, ensuring strong privacy, and setting a new standard for future social networks, prioritizing the freedom of speech and security.

References

[BMHH$^+$21] M. Bernstein, A. Monroy-Hernández, D. Harry, P. André, K. Panovich, and G. Vargas. 4chan and /b/: An analysis of anonymity and ephemerality in a large online community. *ICWSM*, 5(1):50–57, Aug 2021.

[DFMMR16] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. Uncovering the bitcoin blockchain: An analysis of the full users graph. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 537–546, 2016.

[@St24] @StaniKulechov. Ai problem that exists on lens. https://twitter.com/StaniKulechov/status/1787525822913393114, May 2024. Accessed: 2024-05-14.