RARIMO: A PRIVACY-FIRST (ZK) SOCIAL PROTOCOL

WHITEPAPER

May 23, 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 need to be revised to support applications that demand robust privacy and verifiable transparency. Typically, these protocols sacrifice openness for privacy or compromise anonymity for provability, limiting their utility in various contexts.

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	Poor or self-sovereign	Poor or self-sovereign	Self-sovereign
Interactions	Requests/transactions	Requests	Transactions
Authorship	Public	Private	Private + provable
Social graph	Public	Absent	Private (part. knowledge)
Group management	Centralized	Absent	Deterministic

2 Rarimo Protocol

Rarimo is a privacy-first social protocol. It pioneers a new wave of social applications where users can remain anonymous while preserving their historical activities, statements, and identities.

Rarimo ensures user privacy but also maintains a verifiable history to establish social interaction criteria. This capability allows users to interact in an environment where their metadata is protected, yet their historical actions and relationships can be authenticated upon request. It's not a trade-off between anonymity and probability; it's a way to reinforce these properties by combining them.

2.1 A Heart of Private Interactions

A cryptographic commitment is a mechanism that binds specific data while hiding its content. Rarimo utilizes commitments to standardize all statements at the protocol level. This standardization ensures that the structure and update mechanisms of all social trees (introducing them a bit below) are uniform, allowing the creation of efficient inclusion proofs, particularly when aggregated (i.e., when a user needs to demonstrate inclusion within a set of trees).

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 blinding the statement itself. Usually, the $salt$ is random, but for some cases, it must be deterministic (i.e., for achieving uniqueness).

While leaving the commitment, the user creates an irreversible anchor of the statement, keeping it private. If the user wants to use the statement itself, they should perform the following actions:

- 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 of knowledge of statement and salt (two values that were used for commitment construction)
- Prove that the statement satisfies particular criteria (in the simplest case it's statement revealing, but some statements could be provably queried without disclosure)

2.2 Social Forest

Storing poor commitments directly on-chain isn't the most efficient, scalable, and private solution. If the user proves the data from a certain commitment - it reveals the party which left it (transaction sender). It's possible to create proof that the commitment is stored in the whole chain, but it's pretty difficult and inefficient way. So, the proposal is to include these commitments in specific trees, each with its own designation and properties.

Rarimo protocol defines the following tree types 1:

1. **Statement Trees (ST)**. This type allows putting some basic statements into the tree. The main feature of this type is - the absence of hierarchical relations. At the same time, the ability to add new commitments isn't limited by the time. We can specify two sub-types of **ST**:
   (a) **Adjustable Statement Trees (ADST)**. This sub-type allows to set of predefined rules on how commitments could be added to the tree. It means that to put the commitment into the **ADST** tree user must prove the statement itself satisfies rules (partial knowledge).
   (b) **Arbitrary Statement Trees (ARST)**. This sub-type doesn't have a validation for the data being added to the tree. It consumes any commitment and allows to reveal the statement in the future together with its validation rules. Like a permissionless wall where anyone is able to write what they want.
2. **Credential Trees (CT)**. These trees allow the defining of hierarchical relations (one to many) between the issuer and owners of the credential commitments. Each credential tree has an owner (multisig is possible).
3. **Time Trees (TT)**. This type allows the creation of a tree for a particular range of time. These trees can be ST as well as CT; the main difference is the inability to add commitments in the tree after some point in time.

It's always possible to use only one type (**ARST** for example), but by segregating commitments into different tree types according to their properties, users can choose the level of rule enforcement for their commitments, catering to both structured and flexible interaction needs. This

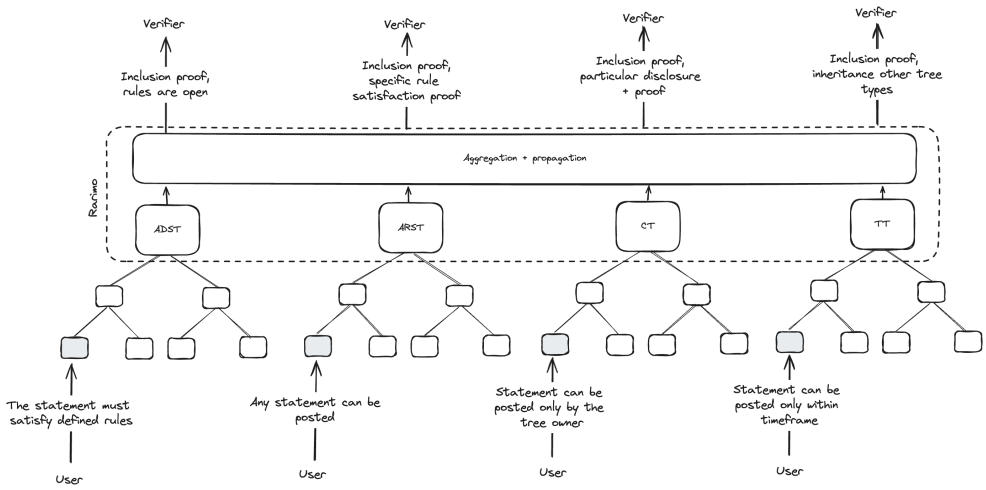

Figure 1: Tree types.

adaptability makes the protocol suitable for a wide range of applications, depending on dynamic requirements.

The social forest's flexible structure also is designed to support future security improvements (new signatures, proving schemes, etc). As new needs and technologies emerge, the protocol can evolve without disrupting existing operations or requiring significant overhauls. Just as real trees grow in accordance with the surrounding climate, so a social forest can exist as a result of a trade-off between efficiency and advanced security.

2.3 Rarimo Core

There is a blockchain layer designated for collecting and timestamping identity events and organizing the social forest—Rarimo Core. This layer is needed to create the social metagraph (see section 3) and propagate it succinctly over connected networks. So, the final social applications can exist on different networks; Rarimo Core allows plugging the single identity layer into the required environment.

Rarimo Core is maintained by a list of validators who can be elected by the Rarimo users. When validators reach a consensus, the metagraph's state is final and can be propagated over other chains. For connected networks, these states are automatically updated by Rarimo oracles, but there is no limitation on connecting new chains or platforms by using their own mechanism for fetching states from Rarimo Core.

2.4 Examples of Tree Types Usage

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

- If the user wants to create a permissionless chat they can use **ARST** tree. Each message will be represented by appropriate commitment. If the user wants to be anonymous, they can use the random salt and signatures for it (and don't care for their storage). If they want to be able to prove the sequence of messages in the future, they should use deterministic salt or the same keypair for signature.
- If the user wants to create a chat only for users that satisfy defined criteria (for example included in some tree) - they have to use **ADST** tree. In this case, for sending the message, the user needs additionally to attach the proof of eligibility. The eligibility criteria are clear - they are defined by the chat creator and verified by the smart contract.
- If the user wants to be an identity provider (for example it represents some authority) and issue some verifiable credentials to their consumers - they should use **CT** tree. This

tree can be updated only by its owner (ARST and ADST tree types don't have owners, but rather initial creators).

- If the user wants to create an event that represents some actions within a particular timeframe (petition signing, periodical check-in for liveness proof, etc.), they have to use **TT** trees.

3 Incognito Social Graph

Obviously, several leaves of the social forest can belong to the same identity, and the identity owner can prove their relation. These relations are bridges between social quorums, and each leaf is an additional criteria that can be recognized by them. It builds a kind of graph that is not only user-centric but also quorum-centric.

We can recognize it as a private social metagraph. The global metagraph, where the user can highlight some zones (relations) and criteria to satisfy certain community entering rules. Examples of such rules are the following:

- Prove the credential and attestation at a particular time (and continuously).
- Prove that some data in the commitment passed the verification defined by a particular dApp or verifier.
- Give/receive attestation to another identity owner (of the whole identity or some parameter).
- Prove that a particular user initiated some actions.
- Prove that specific actions were initiated by the set of identities (group of people), and all/some/at least one satisfies defined requirements.
- Prove that some claims and/or artifacts are connected to one identity.
- Prove that the user is(not) included in specific lists or groups.
- Prove that the sent/received attestation came to/from the identity or service that satisfies some criteria without revealing the identity owner.

We can illustrate these interactions in the following way as in Figure 2. External auditors can see only trees (sometimes only their roots if these trees were built off-chain), but only the owner of knowledge can build relation between some commitments and reveal the part of the graph shaped according to the required community.

4 The road ahead

Implementing the social metagraph introduces several challenges that are critical to address for ensuring the functionality and usability of the system. Here are some of them which Rarimo will solve:

- **Standardization**. Despite the fact that the Rarimo protocol is quite flexible in terms of how statements and commitments can be stored, most applications built on top require standardization and unification of approaches for identity management. So the next step in this direction is connected to the creation of a set of tools that realizes the basic mechanics for social applications. We believe that the majority of interactions (for example, both-side attestations) will be performed the same way for all applications, and for seamlessness and their interaction with each other, these things must be standardized.
- **Advanced and efficient security**. The majority of the user's actions and corresponding proofs will be performed on the user's devices. Rarimo provides tools for generating SNARK proofs for certain types of actions, but the future vector of development is linked to adding more reliable proving schemes and supporting more difficult queries (recursive especially).
- **Recovery difficulties**. Users need to store a lot of data, required for creating proofs for their actions. The problem arises, for example, when the user wants to sync different

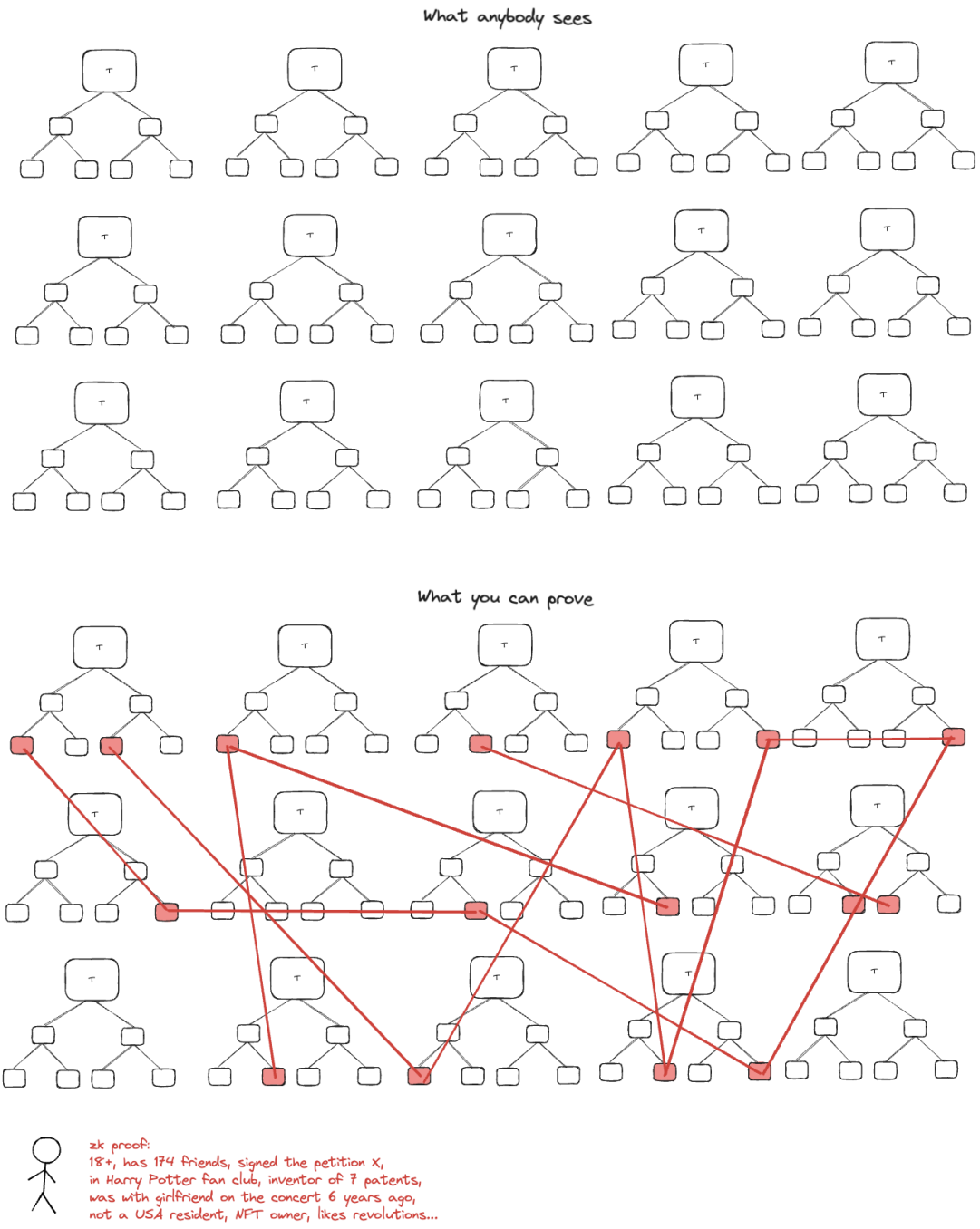

Figure 2: Proofs over the private metagraph.

devices with the same profile data. It means that a lot of data must be able to be backed up and transferred securely. An additional problem is recovering user's access to their profile. If the user loses salt values or parts of statements in open form - they aren't able to create corresponding proofs. From the user experience perspective, it's a very important vector of future development.

- **Calculations top on the metagraph**. Another significant challenge is designing a mechanism that can perform computations or extract insights from the graph without revealing specific events or sensitive information. These requests must be capable of processing encrypted or anonymized data, ensuring that the outcomes of the queries maintain the integrity of the privacy-first approach while still providing valuable results.

Addressing these challenges is essential for the successful implementation and adoption of the Rarimo protocol. By overcoming these obstacles, the protocol can provide a secure and private framework for social interactions and data exchanges in a decentralized environment, paving the way for more user-centric and privacy-preserving applications.

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.