Depo Protocol - High Performance DeFi for Staked Crypto Assets Through Zero Knowledge Homomorphic Smart Contract and Rollup Blockchain

DepoProtocol Team

contact@depo.finance

GitHub: https://github.com/DepoPlatform

Abstract

The Depo Protocol offers innovative, rapid, and efficient execution of decentralized financial transactions for staked crypto assets on the blockchain. This is achieved through a rollup blockchain architecture, ensuring streamlined and speedy execution.

Moreover, the Depo Protocol includes privacy-preserving smart contract capabilities enabled by homomorphic encryption. This feature addresses a major concern for financial institutions and promotes the widespread adoption of decentralized financial transactions for staked crypto assets.

Comprising a rollup blockchain infrastructure and a privacy-preserving smart contract system, Depo Protocol is fully compatible with the Ethereum Virtual Machine (EVM). The Depo Rollup Blockchain boasts a theoretical transaction throughput of over 40,000 transactions per second (TPS), making it suitable for real-world applications.

Furthermore, the Depo Privacy Preserving Smart Contract system offers complete EVM compatibility and portability within the existing Ethereum ecosystem.

Figure 1: Depo = Rollup Blockchain + Homomorphic ZK Smart Contracts for Staked Crypto Assets

Contents

List of Figures

1 Introduction

Staking is a crucial mechanism for securing proof-of-stake blockchain networks such as Ethereum. Participants in the network can operate validator nodes by staking tokens, putting them at risk of being slashed if the node behaves maliciously or proves unreliable. Although many individuals run solo nodes, staking as a service (SaaS) providers allow anyone to stake tokens, exposing them to the same risks while offering potential rewards.

However, staked tokens face limitations—they cannot be freely transacted or used as collateral to generate yield across the decentralized finance (DeFi) ecosystem.

To address this liquidity challenge, liquid staking service providers introduce a solution by minting a new token that represents a claim on the underlying staked asset. This token can then be freely traded or deposited into DeFi protocols. For example, a user can deposit ETH into the Lido staking pool and receive stETH (staked ETH) tokens in return. These stETH tokens can then be deposited into Aave to earn yield. Essentially, liquid staking enhances existing staking systems by unlocking liquidity for staked tokens.

1.1 Liquid Staking

Liquid staking offers the advantages of traditional staking services while also enabling the utilization of staked assets as collateral within the DeFi ecosystem. Providers of liquid staking accept user deposits, stake these tokens on behalf of users, and issue them a receipt in the form of a new token. This new token represents the staked assets and can be redeemed for the original tokens (plus or minus a portion of rewards and penalties). Additionally, this new token can be traded or utilized as collateral in DeFi protocols, thus unlocking the liquidity of the staked assets.

1.2 Benefits and Limitations of Liquid Staking

Advantages of Liquid Staking

1. Enhanced Liquidity: Traditional staking locks tokens in a network like Ethereum, preventing them from being traded or used as collateral. Liquid staking tokens unlock the value of staked tokens, allowing them to be traded and utilized as collateral in DeFi protocols.

2. Composability in DeFi: Representing staked assets as tokens enables their use across a wide range of DeFi protocols, including lending pools and prediction markets.

3. Reward Opportunities: Liquid staking not only provides rewards for verifying transactions, as with traditional staking, but also allows users to earn additional yield across various DeFi protocols.

4. Outsourcing Infrastructure Requirements: Liquid staking providers enable users to participate in staking rewards without the need to manage complex staking infrastructure. Even if users do not possess the minimum 32 ETH required to be a solo validator on the Ethereum network, liquid staking allows them to still share in block rewards.

Risks and Limitations of Liquid Staking

1. Slashing: Users of liquid staking services are exposed to the risk of having their funds slashed if the service provider behaves maliciously or unreliably, as they are essentially outsourcing the maintenance of running a validator node.

2. Exploits: Depositing tokens with a liquid staking service provider exposes the funds to risk if a node operator's private keys are compromised or if the protocol contains smart contract vulnerabilities that could be exploited.

3. Secondary Market Volatility: The price of liquid staking tokens is not pegged to the underlying asset they represent a claim on. While they typically trade at the same price or at a slight discount, they can drop below the price of the underlying asset during liquidity crunches or unexpected events. Additionally, the lower trading volume for staked tokens compared to underlying assets can magnify market shocks, leading to increased volatility in staked tokens.

2 Decentralized Finance of Staked Crypto Assets - Liquid Staking Derivatives

Depo Protocol's Liquid Staking Derivatives aims to bridge the gap between staking rewards and liquidity. In this model, users can deposit their tokens with a liquid staking provider, who then stakes them on their behalf. The provider issues a receipt or token representing the staked assets, which can be traded or used as collateral elsewhere.

One of the primary benefits of liquid staking derivatives is the ability for users to earn staking rewards while retaining liquidity over their assets. By depositing tokens with a liquid staking provider, users can unlock the value of their staked assets and access immediate liquidity. Additionally, liquid staking derivatives allow staked assets to be traded as derivatives, such as futures or options, providing investors with new avenues for financial exposure while still earning staking rewards.

Liquid staking derivatives offer a solution to the illiquidity problem faced by stakers, allowing users to earn staking rewards while retaining access to their staked tokens. This flexibility is particularly valuable for users who need to utilize their tokens for other purposes or require the flexibility to move their funds as needed. For instance, on the Polkadot blockchain, stakers are subject to a 28-day "unbonding period" when wishing to unstake. In scenarios where stakers need to react swiftly to sudden market movements, selling a liquid staking token provides a more expedient option compared to waiting to unstake.

Moreover, liquid staking derivatives enable investors to hedge their exposure to staking rewards. For example, an investor staking Ethereum may be concerned about the volatility of Ethereum's price. By utilizing liquid staking derivatives, the investor can hedge their exposure to Ethereum's price fluctuations while still earning staking rewards. This hedging capability enhances risk management strategies for staking participants.

According to Blockdaemon, approximately $7.5 billion worth of ETH is currently held in liquid staking protocols, representing 20% of the total share of ETH staked in ETH 2.0 contracts. This influx of liquidity into the cryptocurrency markets fosters flexibility, growth, and increased participation in staking activities.

Additionally, liquid staking derivatives offer a more efficient way for users to participate in staking. Instead of individually staking their tokens and managing their own nodes, users can deposit their tokens with a provider who handles the staking process on their behalf. This streamlines the staking process, saving users time and effort, while also reducing barriers to entry for staking participation.

3 Depo Protocol

Depo Protocol provides an end-to-end solution for liquid staking derivatives, and includes: 1) high performance EVM equivalent rollup blockchain as the execution engine 2) homomorphic zero-knowledge privacy preserving smart contract system for secure DeFi transactions.

Homomorphic encryption based privacy preserving smart contract system offers privacy of transaction data which prevents issues such as front running a transaction by third-party validators. Rollup blockchain offers high transaction speed and throughput suitable for high frequency financial transactions. [2]

3.1 Homomorphic Encryption and Privacy Preserving Smart Contract

Ensuring data privacy is a critical concern for smart contracts handling sensitive information. Depo Protocol adopts the ZeeStar system, a language and compiler that allows non-experts to create private smart contracts and perform operations on external data. The ZeeStar language enables developers to specify privacy constraints conveniently using zkay's privacy annotations. The ZeeStar compiler then guarantees the realization of these constraints by combining non-interactive zero-knowledge proofs and additively homomorphic encryption. ZeeStar is practical, as it prepares transactions for our contracts in at most 54.7 seconds, at an average cost of 339,000 gas.[2]

3.2 Rollup Blockchain

Practical usage of RWA DeFi requires user experiences similar to that of traditional financial transactions. To achieve such an end, Depo Protocol employs a rollup blockchain architecture compatible with the EVM smart contract ecosystem. The Depo Rollup Blockchain is built on top of Arbitrum Rollup Stack, which has been proven in real world financial transactions. [1]

Armed with two key innovative features, privacy preserving smart contract based on homomorphic encryption and EVM equivalent rollup blockchain, Depo Protocol brings privacy and speed to real world asset transactions, and provides user experiences meeting real world expectations. [1]

4 Technical Implementation

Overview Depo Protocol consists of two major components:

1. Depo Privacy Smart Contact system - DPSC

2. Depo Rollup Blockchain - DRB

Depo Privacy Smart Contact system - DPSC is built on ZeeStar. DPSC consists of an expressive language to specify and a compiler to automatically enforce data privacy for smart contracts. DPSC not only supports homomorphic addition, but also multiplication for most combinations of owners. This allows expressing complex applications such as oblivious transfer. Furthermore, DPSC can mix homomorphic and non-homomorphic encryption schemes and is provably private.[2]

Depo Rollup Blockchain - DRB is a rollup based public blockchain, and can be deployed either as a layer 2 or layer 3 blockchain depending on how transactions are settled. DRB utilizes the Arbitrum technical stack customized and optimized for Real World Asset based financial transactions. Depo Rollup Blockchain utilizes the DYMO ERC20 token as form of payment for transaction fees. [1]

4.1 Depo Privacy Smart Contact with Homomorphic Encryption

A. Non-interactive Zero-knowledge Proofs A non-interactive zero-knowledge (NIZK) proof enables a prover to convince a verifier that she possesses a secret without disclosing the secret itself. Specifically, she can demonstrate knowledge of a secret witness w that satisfies a given predicate $\phi(w;x)$ for some public value x, without revealing any information about w other than the fact that $\phi(w;x)$ holds. Here, ϕ is referred to as the proof circuit, w is the private input, and x is the public input.

For instance, in a cyclic group G with generator g and $h \in G$, one can prove knowledge of the discrete logarithm z of h with respect to base g using the proof circuit $\phi(z;h)$, which is satisfied if and only if $g^z = h$.

Zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs) are a type of generic NIZK proof construction that supports any arithmetic circuit ϕ and offers constant-cost proof verification proportional to the size of ϕ (plus a typically negligible linear cost in the size of x). Due to their efficient verification costs, zk-SNARKs are commonly utilized on the Ethereum blockchain.[2]

B. Additively Homomorphic Encryption An additively homomorphic encryption scheme enables the addition of plaintexts corresponding to a pair of ciphertexts without requiring knowledge of private keys. Formally, let pk_α and sk_α be the public and private keys of a party α, respectively, and $Enc(x, pk_\alpha, r)$ represent the encryption of plaintext x under pk_α using randomness r. This scheme is additively homomorphic if there exists a function $\oplus$ on ciphertexts such that for all x, y, α, r, r_0:

$$Enc(x, pk_\alpha, r) \oplus Enc(y, pk_\alpha, r_0) = Enc(x+y, pk_\alpha, r_{00})$$

for some r_{00}, where $\oplus$ can be efficiently evaluated without knowledge of sk_α. It's important to note that both arguments to $\oplus$ must be encrypted under the same public key. Typically, additively homomorphic schemes also allow the homomorphic evaluation of subtraction using a function defined analogously.

For instance, the Paillier encryption scheme is additively homomorphic in Z_n (i.e., addition in Eq. (1) is modulo n) for an RSA modulus n, and exponential ElGamal encryption over a group G is additively homomorphic in $Z_{|G|}$, where $|G|$ is the order of G (see App. B).[2]

Privacy Annotations and Types. To facilitate precise and user-friendly specification of privacy constraints, ZeeStar utilizes privacy annotations inspired by zkay. These annotations track ownership of values within a privacy type system: Data types τ (such as integers and booleans) are extended to types of the form $\tau@\alpha$, where α determines the owner of the expression. The value of an expression can only be accessed by its owner. The owner α may be "all" (indicating the value is public) or an expression of type address. Expressions with owner "me" are referred to as self-owned, while those with owner $\alpha \notin \{me, all\}$ are considered foreign.

To prevent implicit information leaks, private expressions with owner α cannot be directly assigned to variables with a different owner $\alpha_0 \neq \alpha$. Instead, developers can use the $reveal(e,a)$ function to explicitly disclose a self-owned expression e to another owner a.

It's important to note that the privacy annotations entail minimal overhead compared to existing non-private smart contract languages such as Solidity. As discussed further, privacy is automatically enforced by ZeeStar's compiler, eliminating the need for developers to manually instantiate cryptographic primitives.[2]

Compilation. ZeeStar compiles the input contract into an executable Ethereum contract that enforces the specified privacy constraints.[2]

In the output contract, values with an owner $\alpha \neq$ "all" are encrypted under the public key of α using an additively homomorphic encryption scheme. Private expressions are precomputed locally (off-chain) by the sender and only published on the blockchain (on-chain) in encrypted form. Expressions revealed to all are additionally published in plaintext.[2]

In essence, any expression involving only public and self-owned variables is computed by the sender as follows: First, decrypt any private input variables. Then, evaluate the expression using the plaintext arguments. Finally, if the expression is private, encrypt the result using the owner's public key.[2]

Leveraging Homomorphic Encryption. As the encryption scheme used by ZeeStar is additively homomorphic, it also permits the evaluation of expressions. First, the sender re-encrypts the plaintext value val under the public key of to to obtain a ciphertext c. Then, the sender computes bal. In the proof circuit ϕ, ZeeStar ensures that c is computed correctly. Interestingly, the operation $\oplus$ is also evaluated within the proof circuit. While not necessary for privacy, this practice leads to reduced on-chain costs. Additionally, as we will discuss shortly, it allows for greater expressivity.[2]

After constructing ϕ, ZeeStar inserts a proof verification statement into the output contract. When calling the transfer function, the sender must generate and provide a NIZK proof for the circuit ϕ as a function argument proof. The public arguments of ϕ are provided as arguments to verify. If verification fails, the transaction is rejected, and the contract state is reverted.[2]

4.2 Depo Rollup Blockchain

The Depo Rollup Blockchain is constructed on the Arbitrum technology stack, which addresses limitations commonly found in layer 1 smart contract systems. Arbitrum introduces a novel approach to overcome these limitations.[1]

Arbitrum contracts are highly cost-effective for verifiers to handle. When participants act in line with incentives, Arbitrum verifiers only need to verify a small number of digital signatures for each contract. Even in cases where parties deviate from their incentives, Arbitrum verifiers can efficiently resolve disputes regarding contract behavior without needing to inspect more than a single instruction execution by the contract.[1]

Additionally, Arbitrum enables contracts to execute privately, disclosing only hashed versions of contract states. Depo utilizes the Arbitrum technology stack as the foundation of its Rollup Blockchain. Moreover, the Depo Protocol customizes and optimizes the Arbitrum stack for real-world asset operations.[1]

Depo Rollup Blockchain is further optimized for RWA DeFi by providing customized middle-ware modules and precompiled smart contacts, including sub-block time data API. [1]

4.3 Depo Protocol - Privacy and High Performance for Liquid Staking Derivatives

Armed with privacy preserving smart contract and rollup blockchain, Depo Protocol solves two of the major hurdles hindering the wide adoption of decentralized finance transactions for liquid staking derivatives, namely data privacy and transaction speed and throughput.

Privacy preserving smart contracts are usually computationally intensive. This translates into high gas consumption and fees in blockchain ecosystem. Depo Rollup Blockchain solves these drawbacks with a highly efficient and low cost EVM compatible ledger.

Together with privacy preserving smart contract and rollup blockchain, Depo Protocol offers community a viable and robust solution to bring liquid staking derivatives and related DeFi transactions to an industry scale.

5 Depo Protocol Economic System and Governance

Depo Rollup Blockchain utilizes the ERC20 token, DepoProtocolToken DEPO, as a payment method for transaction fees. A portion or all of the transaction fees may be distributed to verifiers of transactions. A portion of the transactions fees may also be burned based on community decision.

Community decisions are based on votes by community members. Community members may vote on a proposal in proportion of the number of DepoProtocolToken owned.

6 Conclusion

Depo Protocol is a high performance, low cost, and privacy preserving DeFi system optimized for liquid staking derivatives. Through innovative homomorphic encryption and rollup blockchain technologies, Depo Protocol provides the necessary technological foundation for the wide adoption of liquid staking derivatives and related decentralized financial transactions.

6.1 Future Work

We plan to further strengthen the Depo system by focusing on the following areas:

- Production grade privacy preserving smart contract templates
- Decentralized spot and derivatives exchanges tailored towards liquid staking derivatives
- Liquid staking token price data integration services

6.2 Acknowledgements

We would like to acknowledge 1) ETH and ZeeStar for providing the foundation for Depo Privacy Preserving Smart Contract system. 2) Arbitrum for providing the foundation for Depo Rollup Blockchain.

6.3 Whitepaper Versions

- Depo v. 1.0 – Jan. 2024, initial release

6.4 Code Base

Codebase: `https://github.com/DepoPlatform`

References

[1] Xiaoqi Chen S. Matthew Weinberg Edward W. Felten Harry Kalodner, Steven Goldfeder. Arbitrum: Scalable, private smart contracts. `https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-kalodner.pdf`, 2018.

[2] ROGER BAUMGARTNER MARTIN VECHEV SAMUEL STEFFEN, BENJAMIN BICHSEL. Zeestar: Private smart contracts by homomorphic encryption and zero-knowledge proofs. `https://www.sri.inf.ethz.ch/publications/steffen2022zeestar`, 2022.