Ren
Unstoppable Privacy

Litepaper
v1.2 Q1 2019

A privacy preserving virtual machine powering zero-knowledge financial applications.

Introduction

RenVM

zkTransaction Layer

Interoperability and SwapperD

Dark Pool Layer

The Inception of a New Ecosystem

A Renewed Vision

Our vision for Ren is to enable a world of privacy preserving applications. Ren will power a new kind of trustless distributed computation. One where all applications are run in secret, preserving the privacy of all users and data.

Blockchain technology has proven itself as a viable platform for financial applications. It has promised a future where users control their own data, and developers can build application logic on a trustless distributed computer. However, before decentralized technology can realize its full potential we must first solve the problem of privacy. Without privacy, it is impossible to build more advanced financial applications, or to build general purpose decentralized applications that truly respect the privacy and sovereignty of users and their data.

Ren is an ecosystem for building, deploying, and running general purpose, privacy preserving, applications using zkSNARK and our own newly developed secure multiparty computation protocol. It makes it possible for any kind of application to run in a decentralized, trustless, and fault tolerant environment similar to blockchains but with the distinguishing feature that all application inputs, outputs, and state, remain a secret even to the participants running the network.

Although Ren is capable of powering any kind of application, finance has emerged as a highly valuable use case for blockchains and other trustless networks. As such, Ren will initially focus on supporting three core components to advance the state of private decentralized finance, and to provide a standard economic layer for future applications.

A zero–knowledge transaction layer will make it possible to store and transfer tokens without exposing wallet balances or transaction amounts. »

An interoperability layer will extend zero–knowledge transactions and make it possible to execute trustless swaps between blockchains, and to bridge tokens from one blockchain to another. »

A dark pool layer will provide secret order matching engines that work with orders known only to their owner. This will allow users to open orders without exposing the price or volume of their order to anyone, including the exchange itself. Using zero–knowledge transactions, users can then settle order matches in secret. »

As the ecosystem grows, and the core components are completed, Ren will introduce development tools to support the open development of private applications of any kind, paving the way for an ecosystem of unstoppable privacy.

RenVM

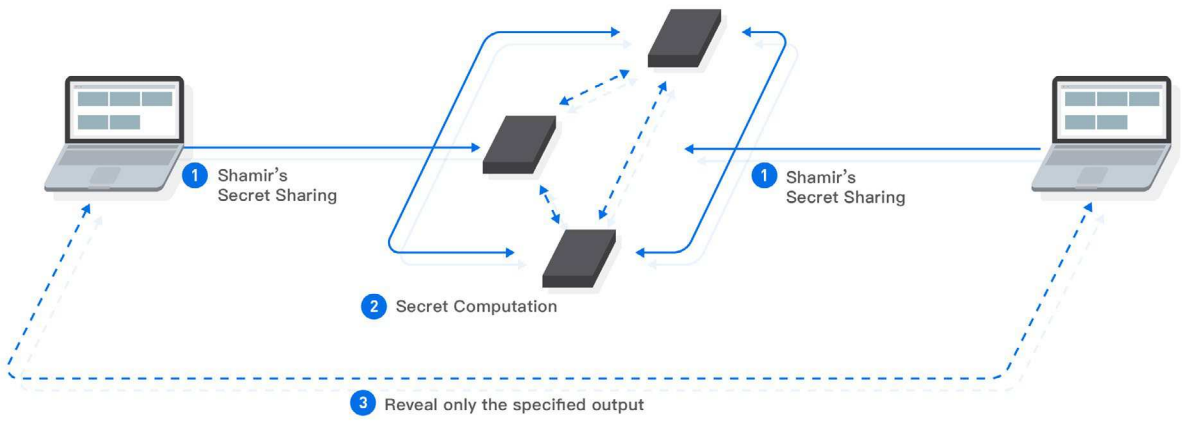
A general use case virtual machine for performing private computations over a decentralized network without revealing the underlying information.

Ren is built on a decentralized virtual machine, RenVM, that leverages zkSNARK and secure multiparty computations to validate proofs and run programs in zero-knowledge. As a general purpose virtual machine, any kind of program can be executed on RenVM while keeping its inputs, outputs, and state, completely secret. Programs on RenVM can also synthesize completely new data, that is not known to anyone, and still be governed by program logic. For example: a program could generate an ECDSA private key that it can use to sign transactions under specific conditions, without the key being known to anyone, including the program itself.

This creates opportunities to rebuild existing financial applications, bringing privacy and Byzantine Fault Tolerance to them, but it also brings about entirely new applications that were not previously possible in decentralized environments. RenVM can be used to replace an ideal trusted third-party: providing absolute secrecy and correctness.

zkCompute and Secure Multiparty Computation

Secure multiparty computation protocols allow untrusted nodes to jointly execute a program without revealing the inputs or outputs of the program, not even to the nodes themselves. This is useful for executing programs that require inputs from multiple users that do not want to share information with one another, without needing to rely on a trusted third-party.



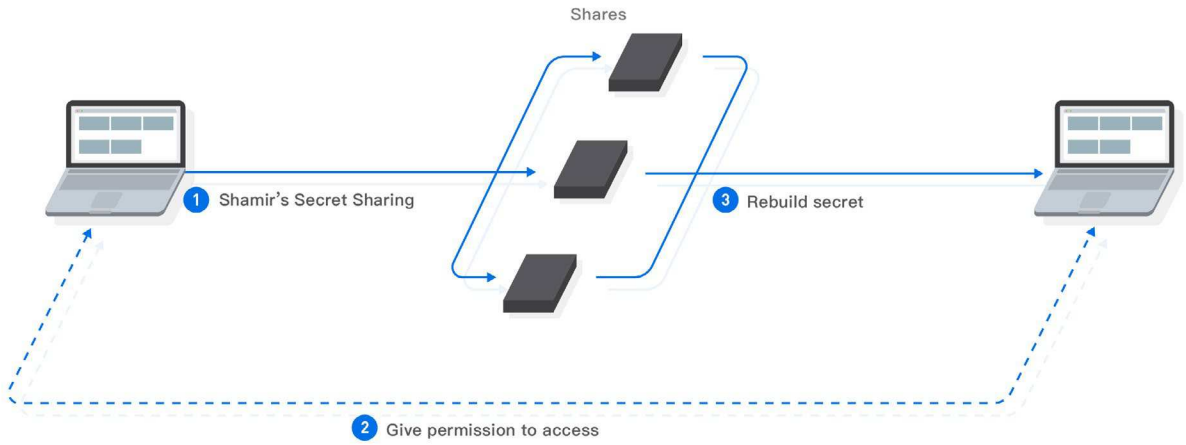
Although secure multiparty computation is general purpose, and can be used to execute any program, modern versions are inefficient and not fault tolerant. They require computationally intensive preprocessing phases that generate large amounts of data, and the entire execution fails whenever one node goes offline. This makes them impractical in decentralized networks where nodes are geographically distributed around the world and cannot be trusted to remain online even when they are operated honestly.

RenVM solves this using a new secure multiparty computation protocol developed by the Ren team. It has a simpler and more efficient preprocessing phase that does not require complex cryptographic primitives and generates less than 1/3rd of the data compared to modern protocols. Security and liveliness is guaranteed as long as less than 1/3rd of the network is malicious. This new secure multiparty computation protocol is at the heart of RenVM, allowing a decentralized network of untrusted nodes to jointly execute general purpose programs in complete secrecy. No data is known to anyone, unless explicitly defined by the computation.

zkStorage and Permission-based Access

Secure multiparty computation is useful for more than just executing programs, it is also effective for storing sensitive information. In RenVM, data is encrypted and distributed using Shamir's secret sharing, a form of informational theoretically secure encryption that splits data into shares. Unless a specific threshold of shares is recovered, the original data cannot be recovered. This prevents anyone, including the nodes running RenVM, from being able to access that data.

RenVM also allows permissions to be defined that can be used to govern access to stored data. This can be done by defining a set of signatures that have read/write access, but it can also be done by defining complex zkCompute programs that evaluate governance rules in secret before granting read/write access. This allows for flexible and powerful permission-based access to secret data without necessarily revealing the identity of accessor.



Using RenVM, sensitive information can be stored securely, that is trustless and fault tolerant. Simple, or complex, rules can be defined making it flexible for all kinds of applications, but particularly well suited for financial tools and enterprises.

The combination of zkStorage, with permission-based access, and zkCompute can be used to build stateful applications that are persistent in RenVM over time. Such stateful applications can keep their state secret, prove properties about their state, and transition their state.

Darknodes

RenVM is powered by a decentralized network of machines, called Darknodes, that contribute their compute power and storage space in exchange for fees. Darknodes are the physical machines that power RenVM, where every machine contributes CPU time for compute power and its disk space for storage. It is important to note that programs executing on RenVM are hidden from the Darknodes that run the virtual machine.

This decentralized network of Darknodes is permissionless, but to prevent the forging of a large number of identities a good behaviour bond of 100,000 REN tokens is required in order to register and run a Darknode. It follows from the maximum supply of 1,000,000,000 REN that there can be at most 10,000 Darknodes. This bond prevents malicious parties from forging a large number of Darknode identities and gaining control over RenVM.

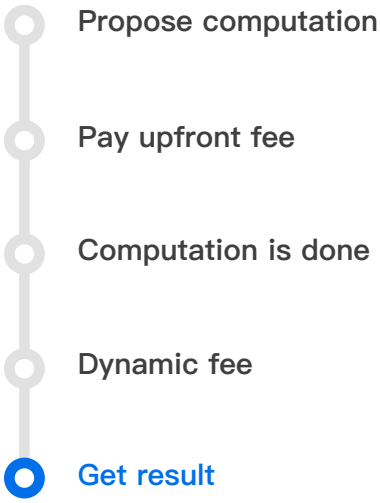
To prevent Darknodes from absconding prior to having their bonds slashed, they are subject to a cooling period before their bond can be refunded, and the identity of the Darknode ultimately deregistered. This cooling period allows others to finalize any outstanding challenges against the Darknode.

Fees and Compensation

Darknode owners are assumed to be rational players that expect financial incentives to contribute resources to RenVM. Therefore, to use RenVM, fees must be paid out to all active Darknodes that are contributing their computational power, storage, and networking.

There are two fees associated with using RenVM: a required upfront fee to compensate Darknodes for the cost of running the computation, and an optional incentivization fee to encourage Darknodes to prioritize the computation.

1. The computation is proposed and an upfront fee is computed.
2. The upfront fee is paid out evenly to all active Darknodes.
3. The computation is executed and the incentivization fee is calculated as one of the outputs of the computation.
4. The incentivization fee is paid out evenly to all active Darknodes.
5. The result of the computation is returned and any state transitions are committed.



The upfront fee must be paid before the proposed computation will be executed by Darknodes, preventing a malicious adversary from consuming Darknode resources without compensation. Because of this stability for fee payments, this primary fee can be tailored to accurately compensate the Darknode owners for the real–world cost of the proposed computation.

The incentivization fee is calculated as a side–effect of the computation, is defined by the computation itself, and can use state produced by the computation (or existing application state). Since the incentivization fee is dynamically calculated, it cannot be known until the computation is completed. Over time, Darknodes will build an expected distribution of rewards for each application and will use the expected distribution to prioritize computations and maximise their revenue. This also has the advantage that the incentivization fee can be delayed or averaged over many computations so that it reveals minimal, or outdated, information about the application. For example: an exchange could define its incentivization fee to be a percentage of transacted volume with a 72 hour delay, encouraging Darknodes to prioritize the exchange without revealing sensitive volume data.

Using this two phase payment mechanism, Darknodes are able to receive stable and reliable compensation for their work done to power RenVM. This payment mechanism also makes it possible to reward Darknodes proportionally for the volume flowing through financial applications, but is flexible enough to incentivize them to run applications that do not necessarily have an obvious analogy to volume. This model is in contrast to other decentralized networks that provide flatter fees, and opens opportunities for higher comparable revenue for Darknodes. This is critical for securing the computation and storage resources used by RenVM especially in the early stages of the network when there are fewer applications and users.

zkTransaction Layer

Allowing for any on-chain Ethereum transaction to be conducted in complete privacy using zk-SNARKs, enabling more sensitive OTC deals and eliminating front running.

Institutions, enterprise businesses, small businesses, and individuals all have well established interests in private balances and transactions. For example: when paying employees it is undesirable for them to be able to trace transactions and discover the salary of other employees. As another example: when purchasing a coffee, it is typically undesirable to allow the cafe to know the wallet balance of the customer. Privacy is critical even in simplest of use cases.

However, in many existing blockchains, all state and all transactions are public. While there are some blockchains that offer privacy natively, like ZCash, providing privacy for blockchains that do not privacy offer natively is an open problem.

Fast and Efficient zkTransactions

One of the unique techniques that is only possible when using secure multiparty computations is the secret generation and storage of private keys. No one, including the participating nodes, can see private keys that are generated in this way. However, modern state-of-the-art secure multiparty computations are not fault tolerant enough to support these kinds of use cases. In modern state-of-the-art secure multiparty computations, one node going offline will cause any operation using the private key to fail and can even result in the loss of the private key. Even when all nodes are operated honestly, nodes in real-world situations often go offline due to bugs, upgrades, network partitions, or even because the node itself is under attack

RenVM uses a novel approach to secure multiparty computations and permits up to 1/3rd of participating nodes going offline (either accidentally, or maliciously). It is also simpler, and faster, than other protocols. As such, RenVM can safely generate and store secret private keys without risking the loss of funds.

Using this ability to generate and store secret private keys, RenVM makes private transactions possible for any blockchain.

1. A private key is secretly generated for the origin blockchain. No one can see this private key and funds sent to it can only be controlled by a respective decentralized and trustless application deployed to RenVM. This will be known as a zkTransactor.
2. Users can deposit balances into the zkTransactor by sending their tokens to the associated public key.
3. zkTransactions can now be made between public addresses by interacting with the zkTransactor. All balances and transfers during this stage are complete hidden.
4. Users can, at any time, withdraw their remaining balance from the zkTransactor. Users that didn't deposit funds, but were sent funds, can also withdraw their balances.

Every blockchain that uses private keys to manage user funds can have a zkTransactor written for it. The majority of blockchains use ECDSA private keys, so support for this signature method will be prioritised.

Compliance and Regulation

As the regulatory landscape for digital assets and distributed ledger technology matures, the need for compliance across multiple jurisdictions is becoming a more apparent need. The zkTransaction layer will support a range of possibilities for proving compliance while minimizing potentially negative repercussions such as the loss of privacy.

Properties, balances, and transaction histories for zkTransactions can be used to cryptographically show that correct behavior was followed without revealing the underlying information. This can be used to provide auditors details regarding specifics about balances and transactions, without revealing information to any other party.

Interoperability and SwapperD

Allowing for any on-chain Ethereum transaction to be conducted in complete privacy using zk-SNARKs, enabling more sensitive OTC deals and eliminating front running.

With the ability to control private keys with decentralized, trustless, and secret application logic RenVM can be used to facilitate more than simple transfers. Different zkTransactors for different blockchains are compatible with each other and can be used to facilitate trustless swapping. This is similar to the way that Ethereum smart contracts can be used to perform trustless swaps between ERC20 tokens, however on RenVM tokens can be from any origin blockchain and the swap amounts are kept completely secret.

For example: Alice has funds deposited into an ETH zkTransactor and Bob has funds deposited funds deposited into a BTC zkTransactor. Alice and Bob are then able to atomically swap between BTC and ETH without any timelocks, or cancellations, and do not expose the amounts swapped.

SwapperD

SwapperD will be a wallet for interacting with RenVM and making zkTransactions secure, simple, and easy to use. It will provide privacy preserving cross-chain swaps with a seamless user experience. SwapperD facilitates cross-chain interoperability to enhance the capability of all products in the Ren ecosystem, as well as providing a universal, user-friendly method for the entire industry to perform trustless swaps between blockchains.

Decentralized Exchange and Application Integrations

SwapperD is an open-source and audited tool, and is designed to be easily integrated into third-party applications and serves as a standard for cross-chain settlements. This paves the way for third-party web applications and servers to integrate with SwapperD and enable users and services to benefit from its interoperable swapping capabilities. Decentralized exchanges in particular can benefit from SwapperD integration by instantly being able to

expand their tokens offerings by utilizing the cross-chain support in SwapperD, and once zkTransactor are completed, offering private settlement to their users.

Blockchain Interoperable Swaps

Initially, SwapperD will perform cross-chain swaps using traditional hash time-locked contracts (HTLCs). Any blockchain that supports HTLCs, such as BTC, LTC, ETH, ZEC, EOS, and countless others, will be able to be supported by SwapperD while it uses HTLCs for cross-chain swapping.

Once zkTransactors have been completed for multiple blockchains, SwapperD will be upgraded to directly use RenVM for more efficient, more powerful, and completely secret cross-chain swaps. This upgrade will also open support for any blockchain, even those that do not support HTLCs. Transactions and balances in SwapperD will become private by default and SwapperD will pay a small incentivization fee to Darknodes ensure that zkTransactions are prioritised and executed quickly.

Dark Pool Layer

Allowing any party to setup and deploy their own dark pool exchange utilizing the hidden order book and privacy technology stack specifically built for Ren dark pools.

RenEx, a dark pool officially supported by the Ren team, demonstrates the use of a hidden order book powered by Darknodes (live trading can be done at https://ren.exchange). This hidden order book will be used to define a standard on the Ren platform for decentralized privacy preserving order books.

Combining this with the zkTransaction layer gives way to the ability for anyone to build a decentralized dark pool that supporting completely private trading. Dark pools are an ideal manifestation of the Ren ecosystem, using all parts of the stack to provide:

Security: Trade without counterparty risk, traders remain in control of their funds.
Transparency: The parameters of the protocol and are immutable and definitive. It is always clear how the market operates and how orders are matched (even though it is 'dark' and the underlying information of each order cannot be seen).
Privacy: Retaining full privacy (orderbook, execution, and settlement) to ensure competitive advantages remain for those who trade.
Fairer implementation of the market: It is not possible to favor any particular party; no entity has an unfair advantage, not even the dark pool operators. The dark pools created on Ren are provably fair by default, unlike parties operating centralized dark pool infrastructure which can take advantage through actions such as peering into the dark pool and front-running their clients.
Greater uptime: Centralized exchanges can suffer extended periods of downtime as a result of factors such as DOS attacks and maintenance. Ren dark pools have no single point of failure and cannot be shut down.

Third-party Dark Pools

Ren will support the implementation and operation of multiple independent dark pools. Each dark pool can define its own set of rules, allowing them to serve and be tailored to operate

in different jurisdictions, with different regulatory requirements, and different settlement options — centralized or decentralized.

Dark pools will be able to use the hidden order book that run on RenVM, leveraging its secret order matching engine, but still maintain control over any KYC/AML requirements that their traders and brokers must meet. Dark pools will also control the supported tokens, and the rules under which new tokens can be supported.

Dark pools will be able to define their method of settlement. Order matching and settlement are inherently decoupled, allowing for arbitrary methods of settlement to be implemented for a chosen dark pool. The zkTransaction layer in Ren will offer dark pools the ability to settle order matches in secret, but it is not required that dark pools use this method. Dark pools on Ren will have the ability to adopt a centralized or decentralized method of operation and settlement.

RenEx
https://ren.exchange

RenEx is an officially supported dark pool built by the Ren team, operating with decentralized custody of tokens and using SwapperD to perform cross-chain settlements. As SwapperD is upgraded to use zkTransactions, cross-chain balances and settlements will become secret on RenEx.

The Inception of a New Ecosystem

The following progression details key components of the Ren ecosystem as they will be completed. Each step marks a significant milestone towards achieving the comprehensive ecosystem that is Ren.

- RenEx Beta
- Swapperd
- 3rd Party Dark Pools
- Zero Knowledge Transactions
- Interoperability Layer
- Open Development
- Unstoppable Privacy

With Ren providing the foundation for a decentralized privacy preserving virtual machine and ecosystem, it will be capable of providing services offered by any current centralized entity in a completely trustless manner. Anyone will be able to build sophisticated financial applications on the blockchain and truly provide general purpose decentralized utility that respects the sovereignty of users and their data.

As the ecosystem grows, and the core components are completed, Ren will introduce development tools to support the open development of private applications of any kind, as well as new products, paving the way for an ecosystem of unstoppable privacy. If you are interested in learning more or building privacy based application check us out on renproject.io for more information.