rev 5b

Provable pTokens

A Provable, portable, and pegged solution for the DeFi ecosystem.

pTokens.io
info@ptokens.io

Abstract

This paper presents a scalable, decentralized solution for DeFi DApps to integrate secure, fully-auditable cross-chain tokens which facilitate the development of multiple financial instruments such as decentralized derivatives.

pTokens.io
info@ptokens.io

Contents

Introduction

Decentralized Finance (DeFi), also known as Open Finance, represents a broad category of financial applications being developed on open, decentralized networks. The objective is to build a multi-faceted financial system, native to cryptocurrencies, that recreates and improves upon the legacy financial system. Being decentralized, DeFi solutions primarily need access to liquidity. In the cryptocurrency space, that liquidity is typically spread thinly across multiple blockchain networks that cannot interoperate.

Currently, the large majority of extant DeFi DApps consists of solutions based on smart-contracts that execute on the Ethereum network. These smart-contracts are only able to consume the native Ethereum token Ether, or tokens built on top of the protocol such as the ERC-20 standard and their ilk.

Therefore, the potential of decentralized finance in terms of liquidity is capped to the one of Ether (ETH) and Ethereum native tokens within the Ethereum ecosystem, to Bitcoin (BTC) within the Bitcoin ecosystem, to EOS and EOS native tokens within the EOS ecosystem and similarly for each and every blockchain protocol. Cross-chain interoperability is a key element to overcome this limitation.

Glossary Of Terms

Provable pTokens

Provable pTokens identify a token that is provable, portable and pegged. A Provable pToken is a token pegged one-to-one to a non-native cryptocurrency, whose issue and redeem is handled by the pTokens technology. For example, a pToken on Ethereum is an Ethereum ERC-20 token pegged one-to-one to a non-Ethereum based cryptocurrency.
For demonstration purposes in this paper, the host blockchain used will be Ethereum and the non-Ethereum cryptocurrency used will be EOS.
The resulting pToken is called pEOS-on-ETH, which for simplicity will be referred to as pEOS when the Host blockchain results obvious.

Native blockchain

The native blockchain is the original blockchain or network to which a cryptocurrency or token is anchored.

EOS

The non-native cryptocurrency used as the example for the pTokens counterparty asset throughout this document. EOS is a delegated proof-of-stake cryptocurrency whose native token EOS cannot currently be used on networks other than EOSIO.

While this document uses EOS as an example, the same logic and process apply for other blockchain networks and assets.

Host blockchain

The host blockchain is the destination blockchain or network hosting a non-native cryptocurrency or token.

Ethereum

The host blockchain used as the example for the pTokenisation mechanism throughout this document. Ethereum is a proof-of-work decentralized network featuring smart contract functionalities.
While this document uses Ethereum as an example, the same logic and process apply for other blockchain networks and assets.

Peg-in process

The peg-in process is the process of tokenisation through which a non-native cryptocurrency is transformed into another digital format compatible with the host blockchain and consequently moved to such a host blockchain.
In the pTokens case, the peg-in process is automated and performed within Trusted Execution Environments.

Peg-out process

The peg-out process is the inverse process of tokenisation through which a tokenised asset is redeemed and returns to its original format.
In the pTokens case, the peg-out process is automated and performed within Trusted Execution Environments.

Trusted Execution Environment (TEE)

A Trusted Execution Environment is a computational environment that is isolated from the main operating system running on a given device. Such isolation is achieved via both software- and hardware-enforced mechanisms.

In general, a TEE runs a small-footprint operating system which exposes a minimal interface to the main operating system running on the device. This smaller footprint reduces the potential attack surfaces of the TEE. Because of this, TEEs can run applications with high-security requirements, such as cryptographic key-management, biometric-authentication, secure payment- processing and DRM. Examples of TEEs are ARM Trustzone-based Secure Elements that is widely used in smartphones, and the more recently released Intel Software Guard Extensions (SGX) Enclaves.

Enclave

Euphemism herein used to describe a trusted execution environment.

Remote Attestation

Remote-attestation is a method by which a device authenticates its hardware and software configuration to a remote host. The goal of remote-attestation is to enable an external party to achieve trust in the integrity of the platform running on the device.

Remote-attestation is often implemented by having the device produce a signed document that attests to the state of the system. The signing is typically performed over a hash digest of the source-code to be attested. The signature is made by a special attesting key that is embedded in the device during its manufacture and over which key the third-party application running on the platform has no control.

This attesting key is known by the manufacturer of the device, that can therefore verify the signed integrity document and return the status of the platform and the code running on it to the party initiating the attestation request.

Blockchain Oracle Architecture

Due to the inherent determinism required by blockchains, it is impossible for them to interface with external datasources. To allow such interfacing, a blockchain-oracle is required to access the eternal datasource, package it into a format recognized by the blockchain, and bake that data onto the chain in the form of a transaction thus making the external data deterministic.

A blockchain-oracle for performing the above has serviced the Ethereum network since 2015 in the form of Oraclize LTD, now Provable Things LTD. Since 2015, Provable Things née Oraclize has expanded their service to include data-fetching from myriad datasources, to implement oracles for blockchains other than Ethereum, as well as to cryptographically harden and decentralize their data-fetching processes through the use of various TEEs.

By providing this same blockchain oracle service running inside trusted execution environments, Provable Things were able to develop and achieve a level of decentralization which aligned with the expectations of users.

For the pTokens use case, the TEE can be considered a blockchain-oracle similar to the aforementioned, whose destination for data is again the Ethereum blockchain, but whose source for this data is the balance of an account on the underlying asset's blockchain.

Multi-Party Computation

Multi-Party Computation (MPC) is a generic cryptographic primitive that enables distributed parties to jointly compute an arbitrary functionality without revealing their own private inputs and outputs.

A fundamental feature of MPC is the ability for the computation to preserve certain security properties, even if some of the parties collude and maliciously attack the protocol.

Unlike traditional cryptographic tasks, where cryptography assures security and integrity of communication or storage, assuming the adversary is an entity outside the system of participants (an eavesdropper on the sender and receiver), the cryptography in this model protects participants' privacy from each other.

In the context of pTokens MPC is used to enable the distributed signing (via a threshold signature scheme) of peg-in and peg-out operations among the network validators.

pTokens network

The pTokens network is the decentralized infrastructure at the basis of the pTokens system. The network is the work-field of validators, actors that operate jointly a multi-party computation via enclaves to generate and manage the key-pairs required to perform the cross-chain movement of assets.

Validator

A validator is an actor within the pTokens network that operates one or more enclaves within one or more TEE-enabled devices. Each validator of the pTokens network coordinates and cooperates with the other validators of the network to perform multi-party computations.

Multisignature Wallet

A cryptocurrency wallet which requires signatures from multiple private keys to authorize transactions.

DApps

A Decentralized Application whose backend logic is powered by a smart-contract or another scripting technique on a blockchain.

DeFi DApps

The umbrella term for the manifold decentralized finance applications spanning various use cases such as decentralized exchanges ("DEXs"), lending and borrowing, derivatives, margin trading and prediction markets.

DeFi Approaches

WBTC - A Federated Approach

WBTC standardizes Bitcoin to the ERC20 format, creating smart-contracts for Bitcoin. This allows bitcoin to take advantage of the greater logic afforded by Ethereum's smart-contracts, expanding the capabilities of the coin beyond simple transfers. The one-to-one pegging of BTC to WBTC is guaranteed by a federated group of actors who manage the private keys behind the bitcoin multi-signature wallet used as the entry point of WBTC generation.
At the time of writing the only custodian partner is BitGo.

Proof of Authority Network (POA)

A POA Network can be both a private or open, public, permissioned blockchain. To reach consensus on the global state, it uses a Proof of Authority consensus algorithm which leverages identity as a form of stake. A group of validators ("authorities") govern the blockchain and validate its transactions and blocks.

The Provable pTokens Approach

The Provable pTokens approach is to decentralize the cross-chain movement of cryptocurrencies via trusted computation and multi-party computation, which together enable secure cross-chain transaction-signing capabilities between two traditionally non-interoperable blockchains. In this manner, two different cryptocurrencies can be exchanged, with their one-to-one peg maintained in a trustless and decentralized fashion, allowing liquidity to flow frictionlessly between chains.

The pTokens Architecture

For demonstration purposes in this paper, the native blockchain used will be EOS and the host blockchain used will be Ethereum. The choice of Ethereum and EOS protocols for this example has been taken given the capability of both networks to support tokenised assets, making a bi-directional cross-chain connection possible. This example showcases the EOS tokenisation process for it to be moved on the Ethereum blockchain. The same process can be applied in the opposite direction for Ether (ETH) and Ethereum-based tokens to be moved on EOS.
The logics and processes presented for the EOS-Ethereum use-case can be applied to any asset and any blockchain.

Overview

The pEOS pToken described in this paper requires the following pieces of infrastructure:

1. An EOS full node.
2. An Ethereum full node.
3. Secure enclaves running inside TEEs.
4. A network of validators that cooperate to jointly generate and manage the private keys for the peg-in/out process.

Both full nodes are required as a source of blocks to be passed to the enclave in order to keep it synchronized with the state of the chains that make up the desired pToken pair. External parties may also submit blocks to the enclave to assist with this synchronization in a decentralized fashion.

The enclave represents the secure sandbox in which private-keys for both corresponding blockchains can be generated, stored and used for the transaction-signing that both funds and burns pTokens. Further logic carried out by the enclave in its secure environment is used to validate incoming blocks and their transactions, ensuring that only valid transactions from either chain can cause their equal and opposite transactional counterpart to be signed by the enclave.

Provable pTokens Journey

The journey from token to pToken (EOS to pEOS in this example) takes place in the following manner:

1) The user makes a deposit of EOS to the EOS pTokens deposit smart-contract, providing their desired destination ETH address in the “memo” field required in an EOS transaction.

2) The block in which the preceding transaction takes place is submitted to the enclave, along with all of its transactions and their actions.

3) The enclave validates the EOS block header along with all the transactions.

4) Once validated, the enclaves locate the pToken transaction sent to the EOS smart-contract, and parse out the EOS amount and the destination ETH address.

5) Using the data from Step 4) the enclaves prepare a transaction to MINT the equal amount of pEOS tokens on the ETH smart-contract.

6) The enclaves perform a multi-party computation to jointly sign the transaction with the derived ETH private key.

7) The enclaves emit the transaction.

8) The transaction is broadcast to the Ethereum blockchain.

9) Once the transaction is mined, newly minted pEOS tokens will now be held by the destination ETH address provided by the user in Step 1).

The converse journey from pEOS to EOS takes place thusly:

1) The user calls the `burn` function in the ETH smart-contract. The function takes two parameters; the amount of pEOS to be burnt, and a `string` representing the desired destination EOS address.

   …

   Steps 2) through 8) are the same as above, with mentions of ETH swapped with EOS

   …

9) Once the transaction is confirmed on the EOS blockchain, the EOS smart-contract will transfer the desired amount of EOS from the EOS deposit address to the destination EOS address supplied by the user in Step 1)

The choice of Ethereum and EOS protocols for this example allow to showcase a bi-directional cross-chain connection. While the process explained until now traces the journey from EOS to pEOS, a bridge in the opposite direction is also possible thanks to the capability of both networks to support tokenised assets.

The inverse journey (ETH to pETH-on-EOS in this example) sees Ethereum as the native blockchain and EOS as the host blockchain and takes place in the following manner:

1) The user makes a deposit of ETH to the ETH pTokens deposit smart-contract, providing their desired destination EOS address.

2) The block in which the preceding transaction takes place is submitted to the enclave, along with all of its transactions and their actions.

3) The enclaves validate the ETH block header along with all the transactions.

4) Once validated, the enclaves locate the pTokens transaction sent to the ETH smart-contract, and parses out the ETH amount and the destination EOS address.

5) Using the data from Step 4) the enclaves prepare a transaction to MINT the equal amount of pETH tokens on the EOS smart-contract.

6) The enclaves jointly sign the transaction with the EOS private key, that is the result of the collective effort of a set of validators who compute such a private key following a Multi-Party Computation technique where each actor taking part in the joint signing leverages a private key sealed within the enclaves.

7) The enclaves emit the transaction.

8) The transaction is broadcast to the EOS blockchain.

9) Once the transaction is mined, newly minted pETH tokens will now be held by the destination EOS address provided by the user in Step 1).

For demonstration purposes in this paper, the two protocols leveraged are Ethereum and EOS. However, the system presents flexible properties that enable an analogous process to be applied for the tokenisation of a variety of cryptocurrencies and tokens.

Implementation details

Generation of EOS and ETH key pair

The enclaves implement secp256k1 elliptic-curve cryptographic primitives (used by both the EOS and ETH protocols) in order to generate asymmetric key-pairs directly inside the encrypted memory of the secure enclave TEEs. Once created, each key is then normally “sealed” in Trusted Execution Environment (for example, SGX) parlance, a process by which

data is encrypted using the private-key baked into the hardware by the manufacturing, before being saved to disc. Only the enclave that sealed the data can decrypt it. This gives strong guarantees as to the safety of the private keys.

The latter process grants each enclave secure access to key-pairs usable for each of the chains it interacts with, allowing it to jointly sign the transaction(s) required in Step 6) of the journey above.

Once created and sealed within each TEE enclave, such key-pairs are leveraged to collectively generate a master key-pair, which is derived from each and every key of the actors participating in the Multi-Party computation. The MPC primitive protects participants' privacy from each other, while enabling the distributed enclave operators to jointly compute the key-pairs that will sign the transaction(s) required to perform the cross-chain movement of assets.

The enclaves act as an extra shield for the system, making it economically and practically inconvenient for a malicious validator to attack the network. Assuming multiple TEE techniques are adopted as a firewall for protecting the key-pairs, for an hack to be successfully performed by the adversary, this would need to bypass multiple layers of protection manoeuvring undiscovered vulnerabilities in a controlled way to break the different safeguards of the relevant Trusted Execution Environments and synchronously replicate such an hack for every enclave operator participating in the multi-party computation.

Block Submission

The enclaves have no networking connectivity in order to reduce their attack surface. Instead, blocks are pushed into the enclave for each of the chains it works with. Anyone is able to submit a block. The enclave validates the block header and its transaction receipts before considering whether to accept a block. If the block has passed this validation step and is the next block the enclave is expecting, the block will be accepted by the enclave, and its transactions parsed. If any transactions relating to pTokens are found, the signature steps will be undertaken. The enclave is agnostic regarding whence a block came, and considers only cryptographic validity when accepting a block.

Block Validation

Incoming ETH blocks are validated by the enclave by first serializing them per the chain in question’s format then hashing that result and comparing it to the block-hash included in the block-header.

Incoming EOS blocks are validated by recovering the public key for the signature associated with the block-header, and checking it against a list of known EOS block validators.

Transaction Validation

In order for the enclave to know the details regarding transactions it needs to sign in order to mint pEOS or move EOS tokens, all the transactions of a block containing a relevant transactions are required to be submitted at the same time as the block in question. Those transactions are then serialized and inserted into a merkle tree in order to discover the root hash of the tree. This root hash can then be compared to the expected transaction root hash contained in the already validated block header.

Scalability

The strong security guarantees offered by TEE enclaves, combined with the flexibility of smart-contracts, allow the Provable pTokens infrastructure to be run by any number enclaves on entirely separate TEE-capable hardware. To allow this, new enclaves can be brought online and their source code be remotely attested for validity before they begin running the pToken code.

Security

The inherent security of TEE enclaves combined with remote-attestation and open-sourced code allow interested parties to independently audit running enclave instances. The publicly available code makes clear every process the enclave is and is not capable of undertaking, whilst remote attestation confirms that that open-source code is indeed what is being executed in the enclave. Because of this, even enclave writers cannot include malicious or hidden entry-points that could compromise enclave private-key security due to the open-source nature of the enclave code.

Multiple Trusted Execution Environment techniques are adopted as a firewall for protecting the key-pairs, acting as a shield composed of multiple layers, each one different and complementary to the others. For an hack to be successfully performed by the adversary (a malicious validator or an external attacker), this would need to bypass multiple layers of protection manoeuvring undiscovered vulnerabilities in a controlled way to break the different safeguards of all the relevant Trusted Execution Environments. The TEE technology acts as an extra shield for the system, making it economically and practically inconvenient for an adversary to compromise the security of the system.

Additionally, such an hack would need to be replicated for every enclave operator participating in the multi-party computation as the key-pairs used to sign the transaction(s) as required by Step 6) of the journey above are not the single key-pairs sealed within each enclave, but rather key-pairs derived from a combination of these.

The use of Multi-Party Computation enables the enclave operators to cooperate and perform the cross-chain movement of assets after they have all verified independently the external blockchains' conditions.

Applications for pTokens

Secure fully-auditable multi-blockchain financial transactions

The blockchain ecosystem is growing and the fragmentation between different networks is an increasing issue for operators and users. Currently, the points of exchange between cryptocurrency assets are often exchange and trading platforms.

These platforms offer great accessibility for users but are lacking in auditability, interoperability and decentralization; every exchange platform has a different trust model, however most are closed source and use a custodian governance model.

Provable pTokens instead are, by design, fully-auditable and the custodian trust model is shifted away from the operator to a TEE. With Provable pTokens, blockchain fragmentation will no longer be a problem because pTokens will form an auditable, secure and decentralized exchange point for multi-blockchain financial transactions.

Enhance external liquidity for ETH DeFi DApps

As previously stated, the DeFi ecosystem is in dire need of two things; increased liquidity, and interoperability in order to grow and overcome Fintech bureaucracy. Instead of leaving that interoperability up to DApp developers to implement on an ad-hoc basis, Provable pTokens will provide that interoperability with no further work required on the DApp developers part.

Indeed, in the case of the pEOS token, every Ethereum DApp becomes accessible to EOS holders and vice-versa. With further pToken integrations, Provable intends to increase DApp interoperability by orders of magnitude between many blockchains, increasing liquidity for all participating chains. This will require no expense in time nor effort for DApp operators, and still preserve a frictionless DeFi experience for DApp users. pToken holders may move their currencies across chains without needing to concern themselves with security and the cross-chain pegging operations at all.

Conclusion

In this paper we have presented a scalable and decentralized solution for secure, fully-auditable multi-blockchain financial transactions. The architecture leverages trusted computing technologies and attestation of code execution to run in secure, tamper-resistant environments whose source is published and third-party auditable. This acts as an extra shield of protection over the network of validators operating multi-party computations to guarantee the pegging of pTokens to their respective underlying assets, making the system decentralized and secure.

For demonstration purposes in this paper, the native blockchain used is EOS and the host blockchain used is Ethereum. The choice of Ethereum and EOS protocols for this example has been taken given the capability of both networks to support tokenised assets, making a bi-directional cross-chain connection possible to showcase. This example explains the EOS tokenisation process for it to be moved on the Ethereum blockchain as well as the process in the opposite direction for Ether (ETH) and Ethereum-based tokens to be moved on EOS.
The logics and processes presented for the Ethereum-EOS use-case can be applied to any asset and any blockchain.