pEOS

pEOS Whitepaper

Private, untraceable transactions on EOS

INTRODUCTION

Would you openly share your entire financial history?

If your answer is no, then you should avoid transacting with non privacy focused cryptocurrencies. Even though internet came with a strong privacy focused culture, we are now living in an age when some of the most successful internet business models involve trading personal information and user insights. In a future when mass adoption of non privacy focused cryptocurrencies occurs, the transaction history of people's finances will be accessible to anyone with the technical capability to reveal it through blockchain analysis. As soon as they manage to match an ID with a cryptocurrency address they can start unravelling the ID's entire transaction history.

Albeit cryptocurrencies entered our lives with the promise of financial freedom, we are at risk of giving up one of the last bits of privacy we still keep safe and away from prying eyes.

Thankfully, there are a lot of projects already trying to upgrade the level of privacy in transactions, but since blockchain technology was still in its early iterations when they were designed, they have several weaknesses that still need to be improved. Our way of doing our part for this cause was to start working on our own modern solution called pEOS.

pEOS is a smart contract implementation of a privacy token, based on the technology that powers the anonymous cryptocurrency Monero, which is capable of running on top of EOSIO software. It allows private and untraceable transactions of, its EOS-based token, pEOS, among EOS users.

Before EOSIO enabled developers to utilize system level languages like C++, the development of highly complex smart contracts like pEOS was almost impossible.

The way to go when building a complex solution was to design and implement a specialized blockchain from scratch. Even though this solution avoids the limitations of smart contract platforms, it introduces new problems:

- Developers have to launch their own blockchain in a fair and secure way
- Community have to continuously provide security for the chain
- Technology behind such chains is bound to limitations of blockchain technology at the time they are being developed, meaning long block times, low throughput and long finality times.
- Independent specialized blockchains like ZCash and Monero have difficulties in getting supported for atomic swaps by other blockchains, due to blockchain interoperability issues, and this makes decentralized exchanges' support improbable.

pEOS avoids these issues by building on EOSIO. Like the industry moved from the era of having to build the operating system along with any software running on a machine, we are now at the point when applications can be built on top of a low level distributed network, like EOSIO based blockchains. It is no longer required to implement the network layer, the consensus systems, etc; instead we can focus on implementing robust systems on top of them, aiming to provide the required functionality and user experience. EOSIO based blockchains are fast with low finality times, that are bound to improve further, and EOS main net is natively supported by decentralized exchanges running on EOSIO. Essentially, pEOS can focus on solving the privacy problem, while in the meantime its main characteristics evolve along with EOSIO software.

Why build pEOS

Even though several popular solutions for private untraceable transactions with cryptocurrencies exist, either by native privacy support with cryptocurrencies like zCash and monero, or through mixer services, the right to privacy is in peril.

Privacy focused cryptocurrencies, being private and enabling untraceable transactions, are in danger of getting deemed as illegal by regulatory bodies in the future. Centralized exchanges in turn will be forced to comply and eventually drop them. Unable to get included to decentralized exchanges, organized markets of these tokens will be hard to manage and remain operational as they will be deemed illegal as well.

Mixer services are centralized and require trusting a centralized actor (the mixer service) to mix coins without tracking inputs and outputs and keeping logs of them. Additionally, since they are centralized, they can be shut down by authorities.

Decentralized exchanges on the other hand, can avoid regulation enforcement, as when implemented and operated properly, they are independent and unstoppable.

Privacy is a fundamental right; decentralized exchanges are the pilar that will ensure that privacy enabling cryptocurrency solutions remain available and liquid to the service of the people.

We are building pEOS to be fast, liquid and private.

"Any society that would give up a little liberty
to gain a little security will deserve neither and lose both."

Benjamin Franklin

HOW IT WORKS

User keys and private addresses

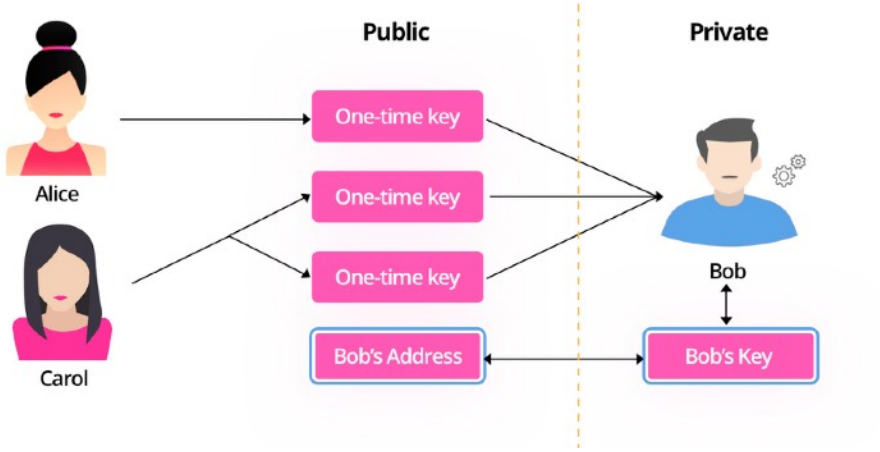
Private tokens can be sent to EOS native accounts like any typical token on the EOS blockchain, but also to private addresses. Native accounts are managed through normal EOS account permissions operations.

However, in order for the system to hold anonymized token at private addresses, it generates two sets of private-public keys, (v, s) and (V, S). s stands for spend key and v for view key.

A user of the pEOS contract has a public address (not related to any of her public keys or accounts on the EOS blockchain), which she can distribute to other users, who can in turn send money to that address.

Tokens will never be transferred in a visible way to that public address. Payments to that address are accomplished through transaction outputs (similar to bitcoin's UTXOs) that pay unique one-time addresses. These addresses are created by doing a Diffie-Hellman exchange so the sender and the receiver both get to the possession of a shared secret.

External users are not able to witness payments to the receiver's public address. However, the receiver can witness and recognize outputs in transactions using her *view key*. She can also proceed to spending them by using her *spend key*.



For cryptography operations contrary to Monero that uses *secp256k1,* pEOS uses the *secp256r1* curve. This is to be in alignment with the native curve used in EOS and in order to be able to support hardware found in modern Android and iOS devices.

Payments to unlink-able addresses

Transactions in the pEOS smart contract contain multiple inputs and outputs similar to bitcoin. However, contrary to bitcoin where payments can be linked to specific addresses that identify the receiver and the sender, in pEOS a transaction does not expose the address of the receiver.

Sending pEOS tokens to a pEOS address does not generate a transaction that visibly identifies the sender, nor the recipient. Tokens are transferred each time at a new address (a public key) that derives from the recipient's address and random data from the sender.

So, assuming that Alice wants to send an untraceable payment to Bob *(Bob has a published address* (V, S)*),* she will generate a random number $r \in [1, l-1]$ and compute an one-time public key:

$$P = \mathcal{H}_s(rV)G + S$$

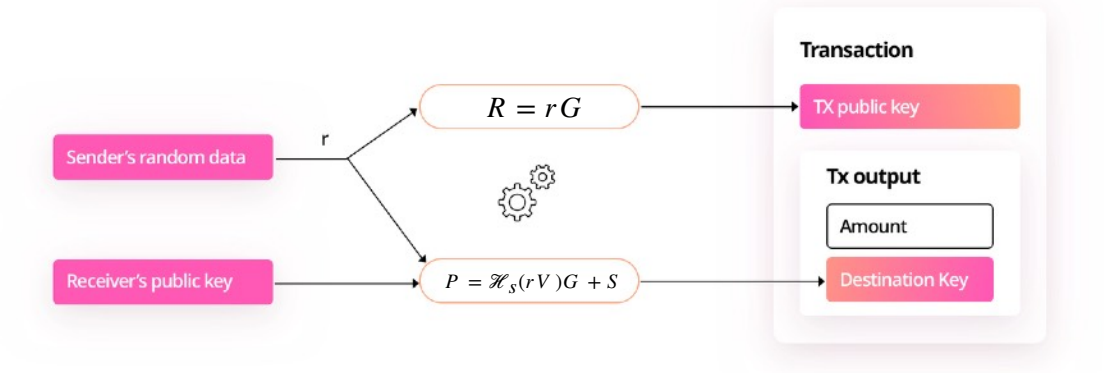
P will be the output address for the transaction. In the transaction she will also include $R = rG$ to allow for Diffie-Hellman exchange with Bob.

On Bob's side, Bob will be able to witness and verify that the output in the transaction can be spent by him. He will compute $P' = \mathcal{H}_s(vR)G + S$ and check if $P' = P$. If equal the payment is to him because:

$$P' = \mathcal{H}_s(vR)G + S = \mathcal{H}_s(vrG)G + S = \mathcal{H}_s(rvG)G + S = \mathcal{H}_s(rV)G + S = P$$

When Bob needs to spend this output he can do so by using his one time private key $p = \mathcal{H}_s(vR) + s$

This way we can see that Bob is the only one that can recognize transactions to his address, and of course he is the only one who can further spend the tokens.



Untraceable transactions

Transactions in pEOS use ring signatures to hide inputs of transactions among inputs from other transactions. It is impossible to verify as a third party what inputs were really used in the transaction.

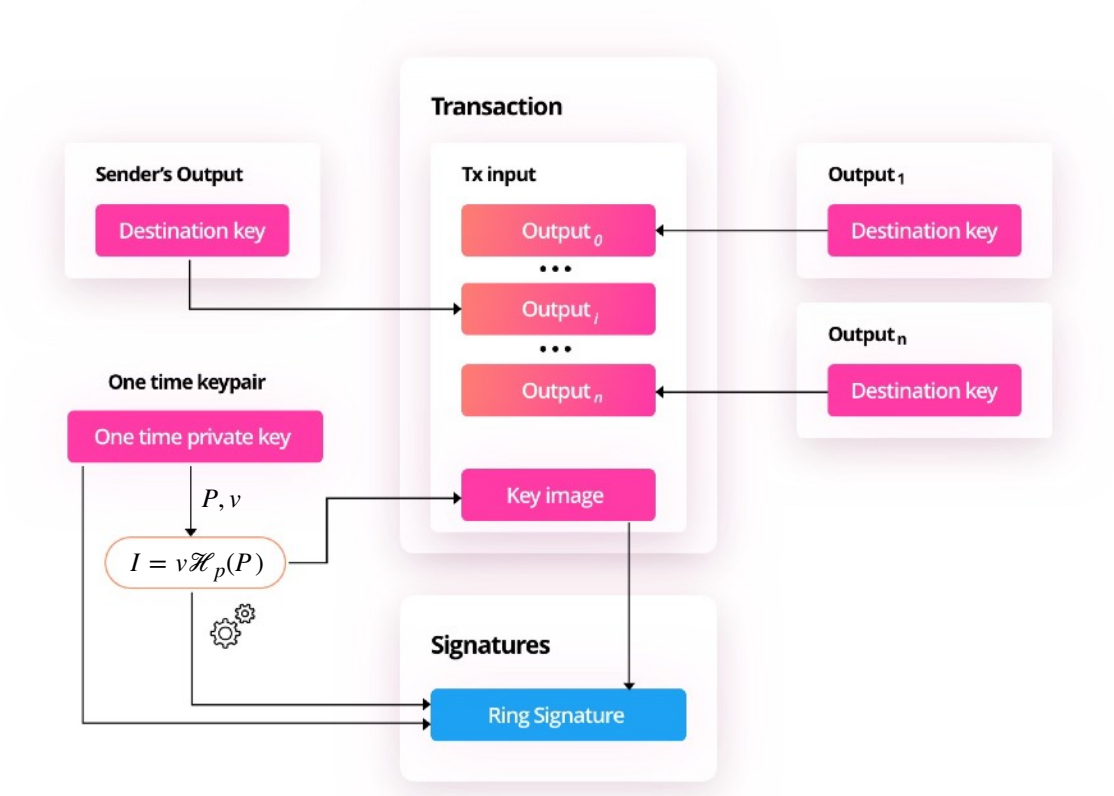
Ring signatures allow for signing a message using several public keys and one private key, with the resulting signature validated by all public keys and the public key of the private key. Created initially as a whistleblower mechanism where the one that signs hides among other public keys he doesn't own the private key. In pEOS we use ring signatures to hide the real spender of an input among n equiprobable spenders.

The level of ambiguity depends on the number n that the sender chooses.

pEOS Transactions

An anonymous and untraceable transaction on pEOS is generated on the user's wallet and broadcasted to the pEOS smart contract for inclusion on the blockchain. So considering the scenario in which Alice wants to send an amount of pEOS tokens to Bob, or more specifically spends an output that was sent to one of the one-time keys she can sign. She can derive the one-time private key for that, using her address' private key, the transaction public key and the output number.

Now in order to send a transaction to Bob, Alice needs to generate a random number (that becomes the transaction's public key). She then uses that, along with the output number and Bob's address' public key to generate his output public key. Alice then, using a ring signature, hides the true output used among foreign output public keys on the system. She signs with her one-time private key, the foreign public keys and the key image of her one-time private key, and appends the resulting ring signature to the transaction.



Hiding amounts

Using the bitcoin UTXO model involves sending transactions that combine or split outputs and recombine them into new outputs, some of which are change returning to the original sender. While we can relay on using coin denominations like Monero did initially, we eventually want to completely hide the amounts on outputs. The pEOS smart contract does not need to know the amounts that are transmitted in a transaction. It only needs to verify that certain conditions are enforced. In particular, the contract needs to verify that the sum of amounts specified in a transaction's input, and sum of amounts in the transaction's outputs are equal.

So if we have a transaction with input amounts $a_1, a_2, \ldots a_n$ and output amounts $b_1, b_2, \ldots, b_m$, then the contract should be able to verify that:

$$\sum_i a_i - \sum_j b_j = 0$$

We are able to verify that without revealing the actual amounts using *Pedersen Commitments*[1]. Pedersen commitments have the property that if $C(a)$ and $C(b)$ are the commitments of a and b respectively, then: $C(a + b) = C(a) + C(b)$

Working with elliptic curve cryptography allows to define a commitment as $C(a) = aG$. Since $(a + b)G = aG + bG$, the commitment is additively homomorphic as required.

However, in order to remove the ability to correlate amounts to their commitments, we add a *blinding factor*. Consider a new EC generator point H such that $H = \gamma G$ with γ being an unknown scalar. Then we define the commitment of value a to be $C(x, a) = xG + aH$, where x is *the blinding factor*. Therefore we disable the possibility to statistically derive a from its commitment.

Suppose now that we have amounts $a_1, a_2, \ldots a_n$. These amounts were outputs in previous transactions and are now being spent. Commitments for those amounts are:

---

[1] Torben Pryds Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing, pages 129–140. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.

$$C_i^a = x_i G + a_i H$$

The new commitment for the same amounts using different blinding factors:

$$C'^a_i = x'_i G + a_i H$$

Subtracting the commitments gives us:

$$C_i^a - C'^a_i = (x_i - x'_i)G = z_i G$$

We can use z_i as a commitment to zero.

The recipient of the transaction should be able to reconstruct the amount commitments, so the blinding factor y and the amount a should be known. For this reason the outputs of a transaction have values *mask* and *amount* defined as:

$$mask_i = y_i + \mathscr{H}_n(\mathscr{H}_n(rV, i))$$

$$\text{amount}_i = b_i + \mathscr{H}_n(\mathscr{H}_n(\mathscr{H}_n(rV, i)))$$

Therefore the receiver will be able to calculate the blinding factor and the amount using rG (included in the transaction) and his view private key.

The blinding factors for inputs and output commitments are selected such that:

$$\sum_i x'_i - \sum_j y_j = 0$$

We can then prove input amounts equal output amounts:

$$\sum_i C'^a_i - \sum_i C_i^b = fH$$

We chose random blinding factors with $x'_m = \sum_i y_i - \sum_{i=1}^{m-1} x'_i$

Interacting with the EOS blockchain

Interaction with the EOS blockchain happens through executing regular actions on the pEOS smart contract. This contract follows the interface of a the native token contract on the platform. Therefore pEOS tokens can be used, transferred and traded like any other token on the platform. Regular wallets can recognize them and provide balance and transfer options.

The smart contract, however, exposes additional functionality that enables the anonymous functionality of the token. More specifically, the contract allows for transfers from normal EOS accounts to *anonymous addresses*, transfers from *anonymous addresses* to *anonymous addresses* and finally transfers from *anonymous addresses* to regular EOS accounts.

To utilize those functions the wallet software needs to be aware of the special functionality of the contract, and be able to construct the special transactions that make the transfers. Additionally, the wallet is safekeeping the keys to the anonymous address the user owns.

The user is only required to store one private key (v, s) and to publish his corresponding public key (V, S). On this public address he is able to receive anonymous transfers and authorize payments from funds belonging to that address. Transfers to that address appear on the chain as unrelated transfers to unrelated one-time addresses.

When the user wants to send tokens, she can choose among the one time addresses that she owns (and only she knows she does), the wallet will then choose n foreign addresses and authorize the payment with a ring signature containing all those foreign addresses.

This transaction will convey no information as to who the sender is, nor as to who the recipient is. Committing this transaction on the EOS platform will require RAM, which should be considered as fees to get your transaction included. The bigger the n chosen, the bigger the fee. RAM usage increases linearly to the number of mixing keys chosen.

The actual pEOS transaction will be encoded as a compact binary serialization of the anonymous transaction data, and passed on to the smart contract.

The PEOS token

Total supply: ~1,250,000,000 PEOS

PEOS is a payment token with privacy features and it will have a fixed supply. This means that no new PEOS tokens will be minted in the future.

The exact number of airdropped tokens will be the determined at the snapshot date, based on the EOS supply, which is estimated to be slightly above one billion tokens. Unclaimed PEOS tokens will be used towards the operating and marketing fund.

Distribution through airdrop

On 15th of February 2019, we will take a snapshot of the EOS distribution which we will use to distribute PEOS tokens with 1:1 ratio.

The token will also be airdropped to known accounts of exchanges, and we will be pushing for distribution to EOS holders within them. We expect the community to push for their favorite exchange to actually claim and distribute the tokens as appropriate. Exchanges that don't want to distribute should not claim the tokens.

50,000,000 pEOS tokens will be used for operations and marketing and the remaining 200,000,000 tokens will be distributed to the founders and the development team.

The pEOS wallet

While pEOS tokens can be transferred among EOS accounts like any other EOS based token, in order for them to perform private untraceable transactions, a special type of wallet is required. Although we expect that various community sourced implementations will be developed, we are planning on developing our own in house solution.

It will be in the form of a native application for OSX, Windows and Linux that will be able to generate your pEOS specific keys, create untraceable pEOS transactions and by connecting with scatter desktop, use it to transmit untraceable pEOS transactions to the network.

PEOS Smart Contracts

thepeostoken will be used to handle the token logic. PEOS source code will be open sourced and with the help of the community and prominent block producers, it will be thoroughly tested in public EOS test nets prior to its release to the main net.

As soon as the pEOS transacts smoothly on the main network, we will lock the contract with an unusable public key, or by assigning the permissions to the ***eosio*** system account.

Roadmap

We will take a snapshot of the EOS distribution on the 15th of February 2019, 00:00 EST, which we will use to distribute approximately 1,000,000,000 PEOS via airdrop at 1:1 ratio. On the 21st of February 2019 we will start the airdrop and EOS account holders that were included in the snapshot will have 2 months to claim their PEOS tokens. At the moment of the airdrop and until the development of the privacy feature is ready, PEOS tokens will be able to transact and get exchanged like any other EOS based token without any specialized functionality.

We anticipate that public testing of the privacy features will start late in Q2, 2019, to be followed by a public release early-mid Q3, 2019. At the moment of launch we will provide the source of the pEOS contract along with the pEOS wallet source for community review.

In the meantime, along with the development we will be reaching out to decentralized exchanges, bloggers and influencers to raise awareness and build a solid community around pEOS.

Next Steps...

If you are a public speaker, content creator, developer, designer or affiliated with decentralized exchanges and you are interested in helping us build the pEOS ecosystem please reach out to us at hello@peos.one.

DISCLAIMER: This whitepaper represents current thinking of the pEOS team and is subject to change without notice. Nothing should be interpreted as a statement of fact or promise to do anything. It is released in order to be a common ground for pEOS understanding.