Pedersen Anonymous Deposits: Commitment Key Packs

NIX Ghost Transactions Brief Technical Overview v1.0

The NIX Developer Team 2018
www.nixplatform.io

Abstract: The introduction of zero-knowledge commitment schemes using Zerocoin Protocol techniques allows for anonymous coin mixing on a blockchain which enables coin history destruction and **full parameter privacy**. At present, Zerocoin is enabled as an intra-layered protocol for individual clients on a P2P blockchain network. However, NIX proposes a solution that expands the isolated zero-knowledge scheme towards a commitment key pack solution that allows third-parties to conduct *direct* zero-knowledge payments, resulting in an **address-less blockchain payment protocol**.

Introduction

As a soft-fork implementation, a new key scheme can be created which manages and tracks Zerocoin Pedersen commitment values for direct public deposits. To understand how this does not compromise any parameter information from peer to peer, we will break down the Zerocoin payment layout:

To create a Zerocoin Mint, several private parameters are used in order to generate the single public parameter that is shared to the network. A random private ECDSA key is created with every single Zerocoin Mint object, in turn a pairing public key (P) is generated. The public key is then RIPEMD160 hashed into a serialized object to create (S). Finally, we then generate a large random number (R) that is then used to compute the Pedersen commitment (C) for the Zerocoin through use of RSA-2048 as the modulus commitment group. The one public parameter that is now serialized and transmitted through the network is the Pedersen Commitment (C).

We are creating a key scheme called **Commitment Key Packs** that will allow packing of Pedersen Commitments to act as one time key formats for Zerocoin Deposits. Through this, we can conduct full blockchain privacy payments without the need of using any group-able addresses on chain and instead **conduct transactions only using zero-knowledge commitment schemes**. A Commitment Key Pack creates a simpler environment for clients to manage one-time payment locations. By default, Zerocoin works as a fixed denomination payment model, and because of this, each individual Commitment Key can only accept one deposit. The purpose for a Commitment Key Pack is to allow a key format to group commit schemes to allow peers to conduct multiple Zerocoin transactions at once.

Flexibility

This privacy model requires a partial interactive payment setup between peers. Because keys are one time usable, providing the same key to multiple payees will not be accepted, therefore a custom Commitment Key Pack should be provided to each payee. Verification of the payment can be monitored in an offline manner, yet ownership of the payment cannot be proven offline.

Scalability

A drawback to this current model include lengthy and unfriendly key formats. Because this payment protocol requires a partial interactive payment setup, the encumbrances of lengthy keys do not pose a major issue, yet this could be mitigated in the future by enabling sender-receiver parameter calculations. This could be done by compromising one of the private parameters such as the random number (R) used to calculate the Pedersen commitment (C) to be calculated instead by the sender. In this case, the receiver only needs to provide a Commitment Key Pack which packages the public key hashed values which are much smaller in length. Another design that could be used to lessen the key length is integration of bulletproofs to introduce reduction of proof sizes which will directly affect the Pedersen commitment size in turn the Commitment Key Pack. Both these solutions are being looked into further.

By design, the Zerocoin spend process is a feeless operation since in practice the Zerocoin should not be split as ideally there is a minimal amount of denominations. Because a Zerocoin spend is feeless, the payment to a new Zerocoin deposit renders no fee which is beneficial for the network and users. Since this operation is feeless, it warrants more users on the network to have balances stored as Zerocoin instead of public UTXO sets which require fees to fulfill payments. This can greatly increase the privacy set of the network.

Design

To allow for flexibility in the key scheme, the design of a Commitment Key Pack reflects the following encoding format:

$\mathrm{Base}_{61}(C_0 + \ldots C_i + \mathrm{0xFFFFFF(eokp)} + \mathrm{CSize}_0\mathrm{(1byte)} + \ldots \mathrm{CSize}_i + \mathrm{checksum(4bytes)})$

A 4 byte eokp (end-of-key-pack) identifier is placed at the end of the grouped commitment keys, followed by respective sizes in single byte increments followed by a 4-byte checksum. This allows keys to be created based on any amount of Commitment packings which allows for flexibility in one time payments, however big they may be. A 10-packed amount is enabled by default, however simple parameter changes can modify it in the base client.

Implementation

Because Commitment Key Packs do not work like standard key pairs, it is necessary to continually scan each block for a direct match. However, no strenuous calculations are made to determine whether or not a payment has been successfully transferred; the check for ownership can be made in O(1) time which in turn adds negligible time to the blockchain sync process.

When a localized Zerocoin is created currently, the Zerocoin data is written to the wallet database which is then used to provide the ZK proof on Zerocoin spend verification. However, Commitment Key Packs differ in that the ZK proof information is instead stored on a separate wallet database that is used to either generate packs for payments, or to scan each block or transaction to verify ownership.

With Commitment Key Packs being an interactive payment model, simple SPV wallets can be dedicated to hosting the one-time keys associated with these payments. To determine if an existent paid Commitment Key Pack is eligible to redeem on the blockchain, only two factors must be known, the public commitment scheme, and the serialized private coin information. With these two parameters, a light wallet does not need to sync with the entire blockchain, and instead, only needs to create a data structure that holds total public and private redeemed commitment schemes on chain. This can be an extremely light performance based process, and broadcasting payments to the network does not require pre-existing blockchain history.

Benefits

There are many factors that could potentially break a user’s privacy on a single blockchain. Commitment Key Packs create an environment of privacy that cannot be broken by any external parameter.

One privacy failure could be IP linking through node transaction transmission, which can help group and identify transactions made by certain nodes to one owner. A work-around for this would to be to use Tor/VPN and/or dandelion. The issue here though comes when a transaction is funded from multiple change addresses, linking the owner and previous blockchain history rendering the Tor/VPN and/or dandelion useless. To be almost 100% private on all ends of a transaction, a user would always need to make sure proper networking and blockchain privacy is managed; this could add inconveniences to the process of transacting on a specific blockchain network.

With Commitment Key Packs, there is no need for any networking privacy since transactions cannot be linked to public any UTXO sets. Theoretically, if a user only conducts transactions using Commitment Key Packs, there could be no way of compromising his/her history.

Conclusion

We have proposed a full parameter private address-less blockchain payment protocol which can be integrated through a simple soft-fork. The benefits that this payment model can offer, outweigh any encumbrances in its current state. This protocol not only enhances user privacy on a single blockchain, it also simplifies any layer-2 solutions utilizing privacy mechanisms.

References

[1] I. Miers, C. Garman, M. Green, A. D. Rubin, “Zerocoin: Anonymous Distributed E-Cash from Bitcoin”, http://zerocoin.org/media/pdf/ZerocoinOakland.pdf, 2013.