PoSA v3.0

A trustless, anonymous transaction system for CloakCoin.

27th April 2015

1. Abstract

CloakCoin is a cryptocurrency designed to facilitate trustless and anonymous, decentralized transfers with PoSA (Proof-of-Stake-Anonymous transfers), and secure, anonymous and decentralized marketplace trading with OneMarket.
Cloak is a dual PoW/PoS (Proof of Work, Proof of Stake) coin, which is now in the Proof-of-Stake (interest bearing) stage.

PoSA3 is CloakCoin's trustless, anonymous payment system, that forms the basis of future development and provides the underlying transaction system for the decentralized Cloak 'OneMarket' marketplace.

Privacy today is perhaps more important than ever. The thundering pace of technological advancement has rapidly broadened our horizons and connected the world like never before. Thanks to Bitcoin's introduction in 2009, cryptocurrency is steadily moving into the mainstream and we can now transfer digital currency securely across the globe in an instant, using the power of the blockchain. As cryptocurrency adoption becomes more widespread, increased regulation is inevitable. It remains to be seen what form this regulation will take, but many are concerned it may be overly draconian and designed to stifle some of the more libertarian aspects of cryptocurrency.

Due to Bitcoin's quest for ubiquity, anonymity for users has never been a priority for the core Bitcoin developers. This has led to the emergence of a number of privacy-centric, alternative cryptocurrencies, designed with anonymity in mind. There are many different methodologies and approaches to solving the problem of anonymous transactions, and in this document we present a full overview of CloakCoin's PoSA 3 system.

PoSA3 is at heart a decentralized, off-blockchain mixing service which allows users on the CloakCoin network to transmit Cloak anonymously to each other. It has been designed to ensure the mixing process is both trustless and anonymous. This ensures a user's Cloak coins are kept safe during transfer and that the sender and receiver cannot be tied or associated. Cloak coins are never transferred to an intermediate party during Cloaking, so coins remain safe. We have also worked hard to ensure the PoSA design rewards users who assist in Cloaking transfers and will continue to improve the process and further incentivize active participants. Anyone with Cloak coins can participate in Cloaking operations, which allows them to leave their wallet running in Staking/ Cloaking mode to allow it to passively assist in Cloaking and earn significant rewards.

2. PoSA v3.0 Overview

PoSA3 is the third iteration of Cloak’s trustless and anonymous payment system. PoSA transactions are ‘cloaked’ by other users, who receive a reward for their assistance. The other users provide inputs and outputs to the PoSA transaction making it impossible to determine the true source and destination of the cloak transfer. All PoSA messages on the network are hashed and encrypted for the recipient using CloakShield to ensure data security and integrity. Please see Section 3 – 'CloakShield' for more details.

2.1. The PoSA Process (for PoSA enabled nodes)

PoSA Announcements

PoSA nodes communicate over the Cloak network and a node will keep track of other active PoSA nodes. PoSA Announcement Broadcasts alert other PoSA nodes of our public session key and current PoSA cloaking balance.

PoSA Cloaking Requests

When a user wishes to send a Cloaked PoSA transaction, they elect a series of PoSA nodes (with a high enough PoSA balance) and request their assistance in cloaking. A PoSA node can choose to assist in cloaking and send an acceptance response to the requester to indicate this. If a PoSA node declines to participate in cloaking or does not respond in a timely manner, an alternate PoSA node is elected and contacted. DDoS (distributed denial of service) protection will blacklist any misbehaving nodes for the remainder of the session. A node is deemed to be misbehaving if it repeatedly refuses to sign a PoSA transaction or refuses to relay PoSA messages. PoSA cloaking nodes use an Elliptic Curve Diffie Hellman key exchange (ECDH) to derive a shared secret with the PoSA initiating node, which is used to generate a shared secret key for symmetric RSA-256 data encryption between a cloaking node and the sender node.

PoSA Cloaking Acceptance

When a PoSA node accepts a 'cloaking' request, it provides a list of transaction inputs and outputs to be used for the PoSA transaction. Input amounts provided by a cloaking node must be greater or equal to the PoSA send amount (plus any fees). Outputs are carefully selected so that they match the true output of the PoSA transaction as closely as possible. If the PoSA output address has not previously been used, a new change address is generated by the 'Cloaker'. If the PoSA output address has previously received funds, an existing address with similar activity is chosen by the 'Cloaker' to return their input funds and receive the PoSA 'cloaking' reward.

The 'Cloaked' PoSA Transaction

The PoSA Sender constructs a 'cloaked' transaction using the inputs and outputs provided by the PoSA Cloaker nodes. The PoSA Sender then adds their own inputs and outputs to the transaction, before shuffling all transaction inputs and outputs to facilitate 'cloaking'. The 'cloaked' transaction is then encrypted and sent (using CloakShield) to each participating Cloaker. Cloaker nodes check the transaction to ensure the inputs and outputs they supplied are present in the 'cloaked' transaction and that one or more of their outputs has also been rewarded with sufficient fees. If the transaction checks are passed, the transaction is signed *(SIGHASH_ALL+SIGHASH_ANYONECANPAY)*, encrypted and relayed back to the PoSA Sender. Once all PoSA Cloakers have signed the transaction, the PoSA Sender confirms the signed transaction is valid and signs it. The 'Cloaked' transaction is then ready for submission to the Cloak network.

2.2.1. Tracking PoSA Cloaking Nodes

PoSA enabled nodes on the Cloak network broadcast announcements to other nodes. These PoSA announcements contain the public ec-key ID of the node and the currently available balance for PoSA cloaking operations. Nodes maintain a list of other active PoSA nodes on the network so that they can communicate for cloaking purposes. Nodes IDs are generated on a session-by-session basis; restarting the client will refresh the current ID.

1. Each wallet creates a public/secret (secp256k1) key pair for the session at start-up.
2. The wallet announces its public key and Cloaking balance for the session periodically to other nodes on the Cloak network.
3. Nodes keep track of other active PoSA Cloaking nodes and can communicate with them directly or indirectly (via CloakShield Onion Routing).

2.2.2. Initiating a PoSA Transaction

Alice wishes to send 10 Cloak to Catherine using 1 mixer:

1. Alice broadcasts a PoSA request to the network, containing her public PoSA session key and the amount of cloak she wishes to send. Her request is securely routed through a series of PoSA nodes to mask the originator.
2. Bob has ‘Cloaking Mode’ enabled and creates a secure CloakShield encryption channel for secure communication with Alice. Bob then constructs a PoSA response packet and sends it securely to Alice. The response contains a list of Bob’s inputs and outputs that Alice will use to ‘Cloak’ her transaction.
3. Alice decrypts and processes Bob’s PoSA response and creates a PoSA transaction using her own inputs and outputs mixed with Bob’s inputs and outputs. This is encrypted and sent to Bob for signing.
4. Bob decrypts the PoSA transaction and performs a number of integrity checks on the transaction to ensure the inputs and outputs he supplied have been used correctly and that he has been rewarded sufficiently. If the PoSA transaction passes the tests, Bob signs it, encrypts it and transmits it to Alice.
5. Alice performs further checks on the signed transaction before signing it herself. The transaction is then submitting it to the network (securely routed through PoSA nodes) for inclusion in a block.
6. When the transaction is finalized, Catherine will receive the funds from Alice and Bob will receive a ‘Cloaking’ reward for assisting in the PoSA transaction.
7. Due to Bob’s inputs and outputs mirroring Alice’s, it is not possible to ascertain the true sender and recipient of the PoSA transaction.

3. CloakShield

CloakShield provides secure communications between nodes on the Cloak network using symmetric RSA encryption backed by an Elliptic Curve Diffie Hellman key exchange (ECDH). This allows nodes to exchange data securely, providing protection from snoopers (man in the middle) and imposters (sybil attack). CloakShield is designed to secure both PoSA and OneMarket, and will ensure your data stays as private as possible.

CloakShield allows the encrypted sending of data to one or more recipients.
When sending to a single recipient, the payload is RSA encrypted using the ECDH shared secret.
When sending to multiple recipients, the payload is encrypted using a one-time key and the key is then encrypted for each recipient using the ECDH/RSA method.

Generating a shared encryption key

In order for Alice and Bob to communicate securely, they must agree on a shared encryption key. CloakShield uses ECDH to accomplish this:

- Alice has PoSA private key **dA** and PoSA public key **QA=dAG** (where G is the generator for the elliptic-curve). Bob has PoSA private key **dB** and PoSA public key **QB=dBG**.
- Alice has Bob's PoSA public key **dB** from the PoSA announcements he sends to the network to announce his availability for cloaking assistance. She uses her private key **dA** and Bob's public key **QB** to calculate shared secret **dAQB=dAdBG** (ECDH_compute_key in OpenSSL).
- Alice then creates a SHA256 hash of the secret and passes the hash to the OpenSSL EVP_BytesToKey method in order to derive an encryption key and IV, which will be used to encrypt data for Bob (using symmetric RSA encryption).
- Alice is now able to create CloakShield secured messages for Bob.
  When Bob receives a Cloak Shielded message from Alice, he reads Alice's public key from the message header and generates the same shared secret key as Alice, as per the steps above (with his secret key, instead of Alice's). The Cloak wallet maintains a list of active CloakShield keys and will check the list for an existing CloakShield key before generating one.

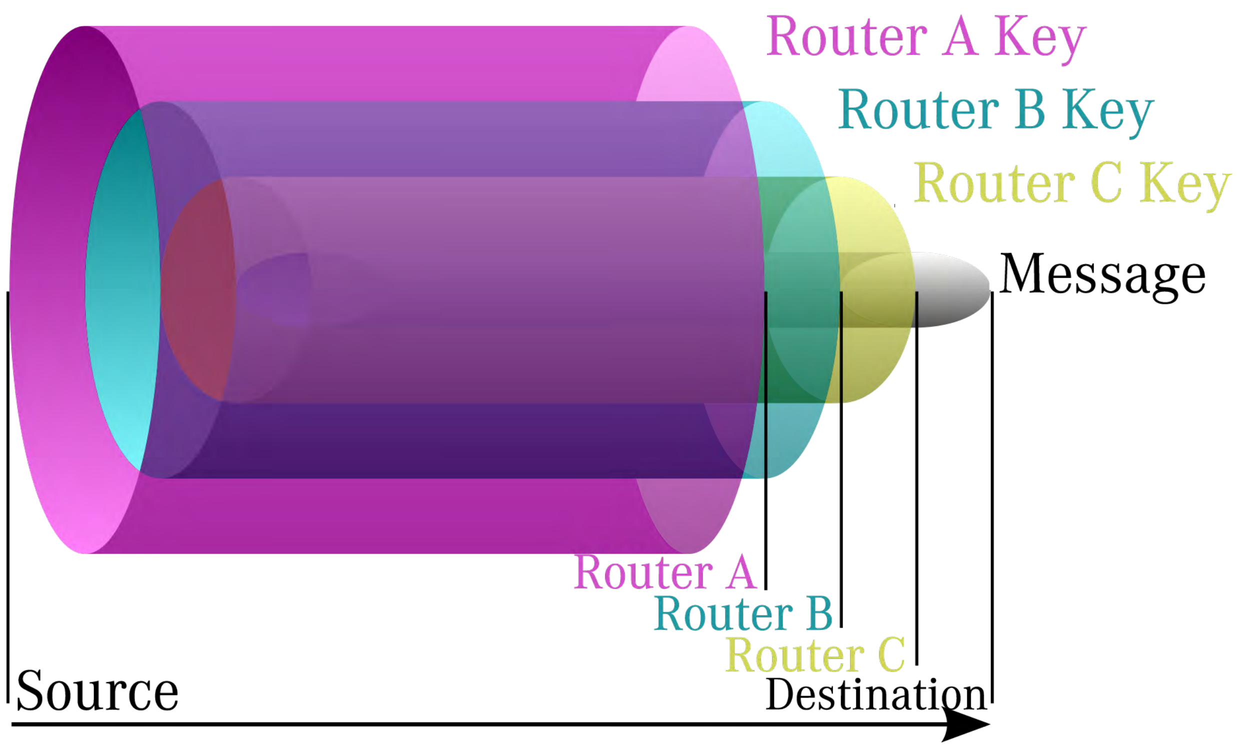
CloakShield Data

CloakShield allows any Cloak data objects to be serialized and transmitted securely to one or more recipients. A CloakShield data packet-header contains the sender's PoSA public key and the public keys hashes of the recipients.

CloakShield headers contain a verification hash which is generated using the sender's public key and the raw unencrypted data. This hash is verified during decryption of CloakShield data to ensure that the recipient info in the header matches the encryption key, and that the data has not been altered.

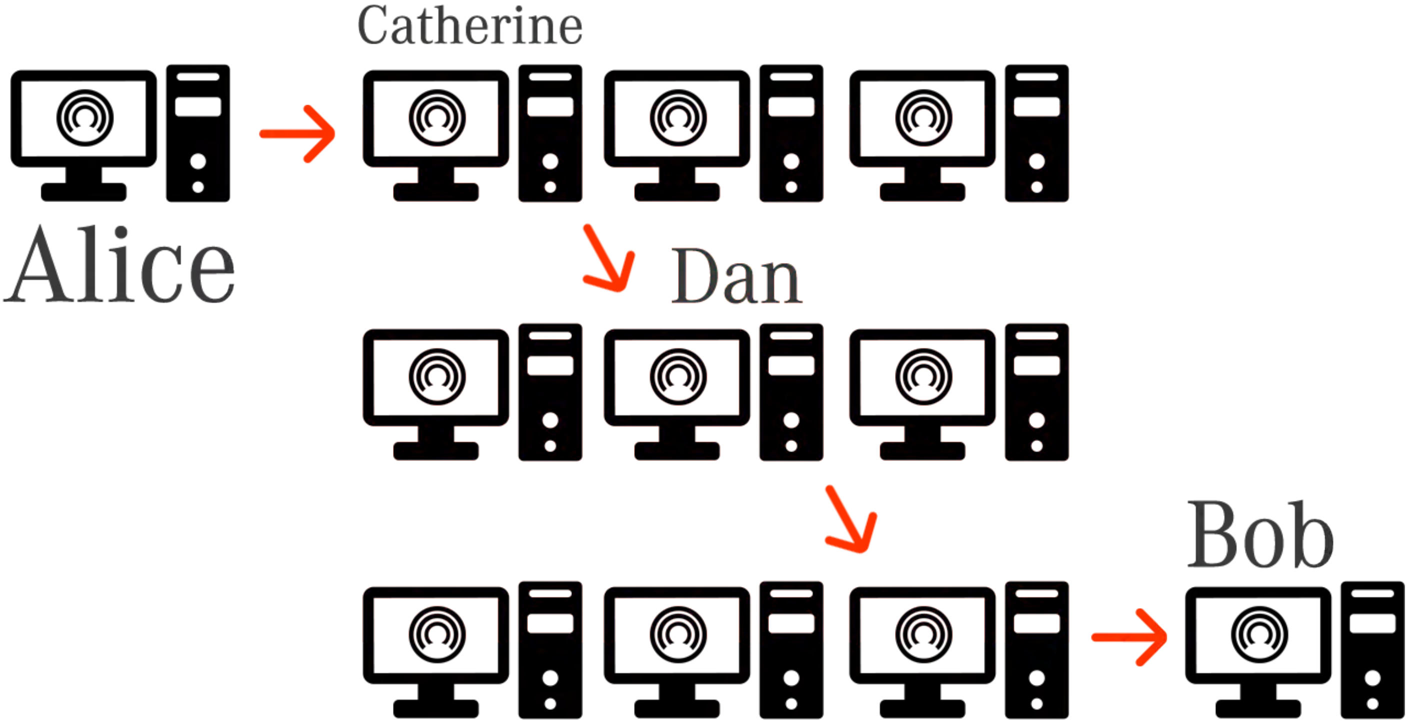
CloakShield Onion Routing

Onion routing is a technique (used by TOR) for anonymous communication over a computer network. In an onion network, messages are encapsulated in layers of encryption, analogous to layers of an onion. The encrypted data is transmitted through a series of network nodes called onion routers, each of which "peels" away a single layer, uncovering the data's next destination. When the final layer is decrypted, the message arrives at its destination. The sender remains anonymous because each intermediary knows only the location of the immediately preceding and following nodes.



Onion Routing Analogy

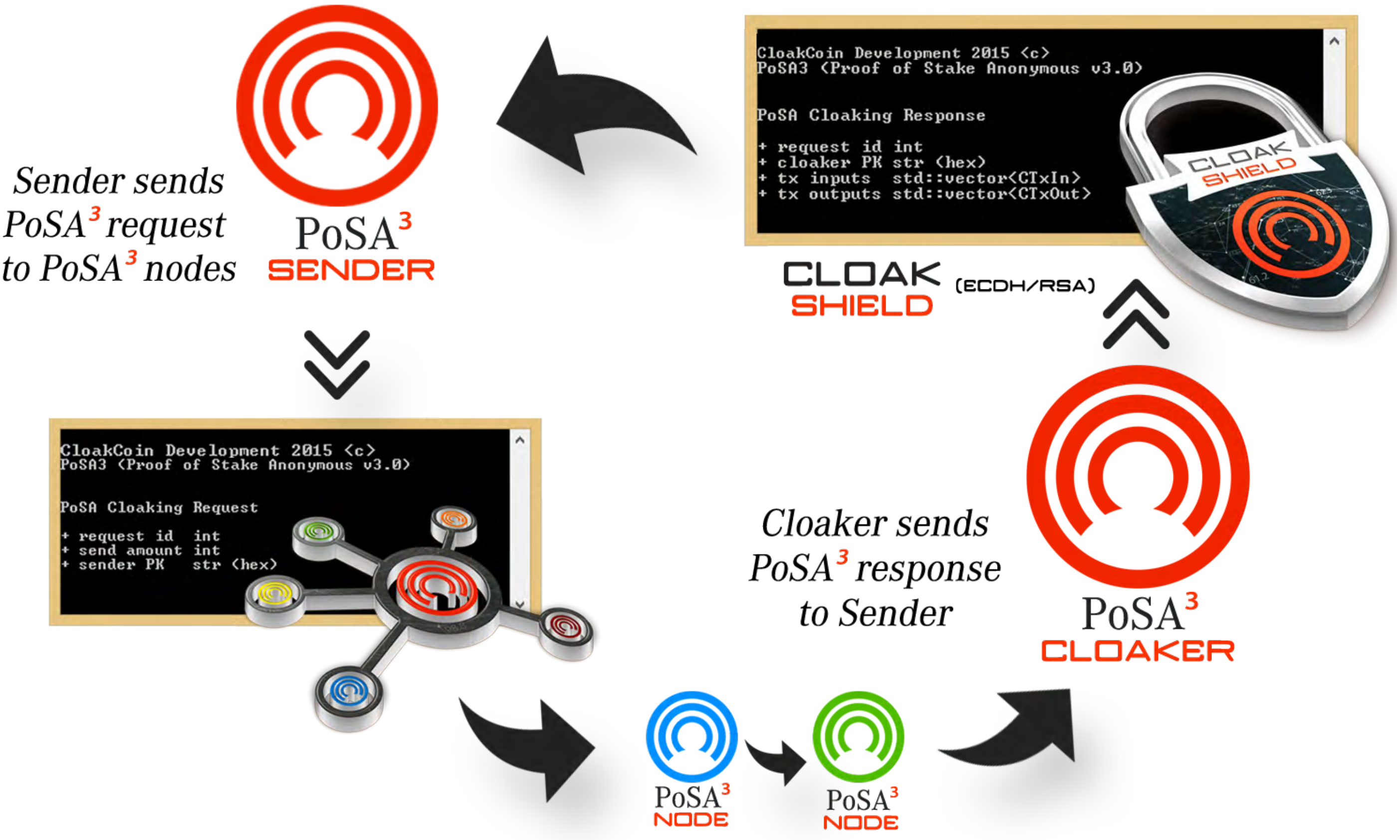
The addition of 'onion routing' functionality to the PoSA network (using CloakShield) allows nodes to communicate indirectly to circumvent traffic analysis. This hampers attempts at determining which nodes are communicating with each other or which nodes submitting transactions to the CloakCoin network. When a PoSA node wishes to communicate with another PoSA node it selects a number of other PoSA nodes to act as relays for the communication. Each encrypted layer can only be decrypted by the intended relay [for which the specific layer was encrypted]. After decrypting a layer, the relay passes the data to the next relay node. This routing continues it until the data reaches its intended recipient and all layers have been decrypted in turn by the selected relay nodes. Due to the self-contained nature of the PoSA network, exit nodes are not required and CloakShield ensures there is no risk of a relay node reading or altering the encrypted data



Alice onion routing a message to Bob via Catherine and Dan
(Catherine and Dan are arbitrarily selected PoSA nodes used for illustration)

4. PoSA Illustrated (high-level overview)

Step 1 *- Request Cloaking: PoSA Request from Sender and Response from Cloaker*



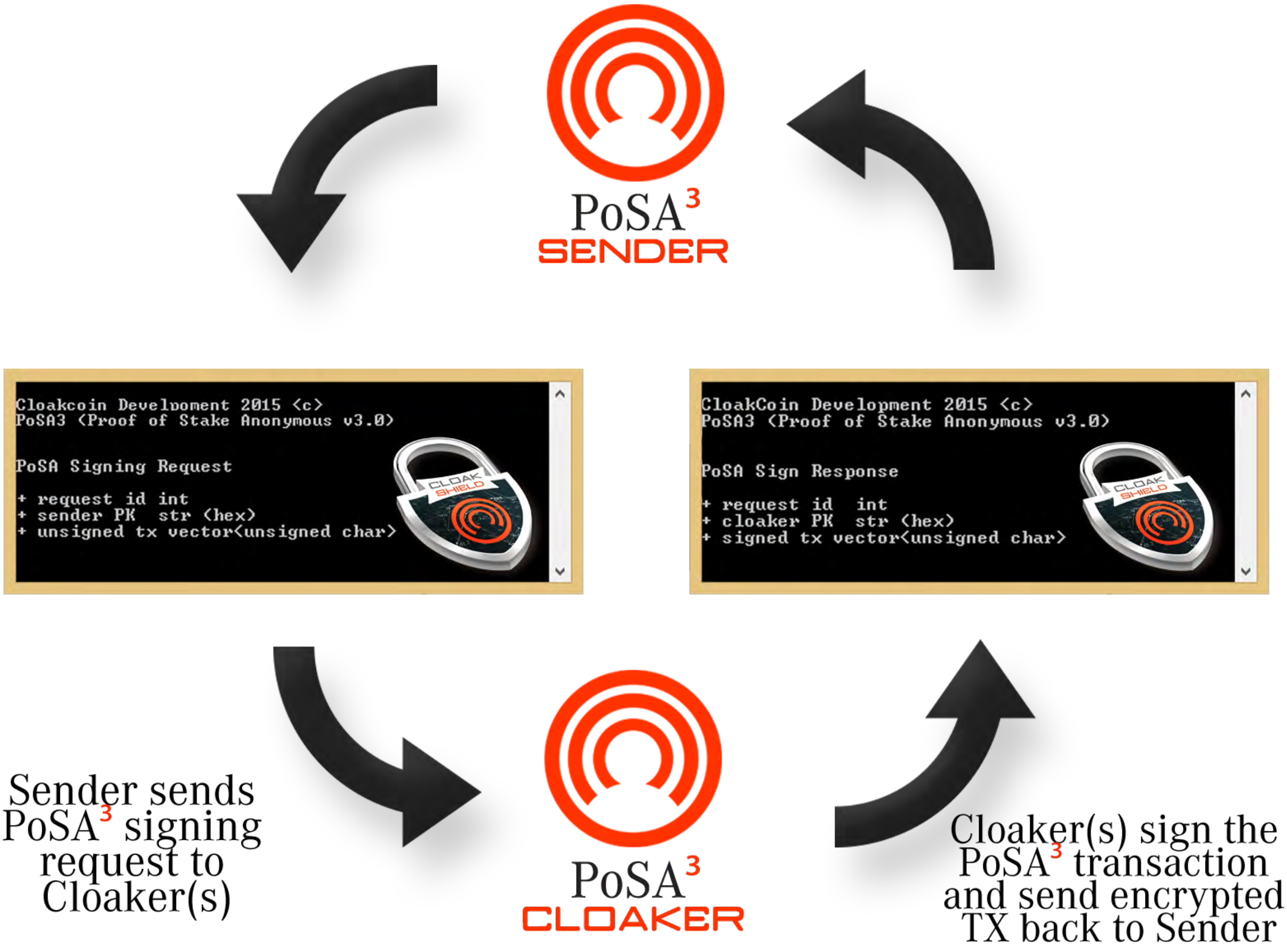
Sender sends PoSA^3 request to PoSA^3 nodes
PoSA^3
SENDER

```
CloakCoin Development 2015 <c>
PoSA3 <Proof of Stake Anonymous v3.0>

PoSA Cloaking Request

+ request id  int
+ send amount int
+ sender PK   str <hex>
```

PoSA^3
NODE
PoSA^3
NODE
Cloaker sends PoSA^3 response to Sender
PoSA^3
CLOAKER
CLOAK
SHIELD
(ECDH/RSA)
CLOAK
SHIELD

```
CloakCoin Development 2015 <c>
PoSA3 <Proof of Stake Anonymous v3.0>

PoSA Cloaking Response

+ request id int
+ cloaker PK str <hex>
+ tx inputs  std::vector<CTxIn>
+ tx outputs std::vector<CTxOut>
```

Step 2 *- PoSA TX Signing: PoSA Sign Request from Sender and PoSA Sign Response from Cloaker*



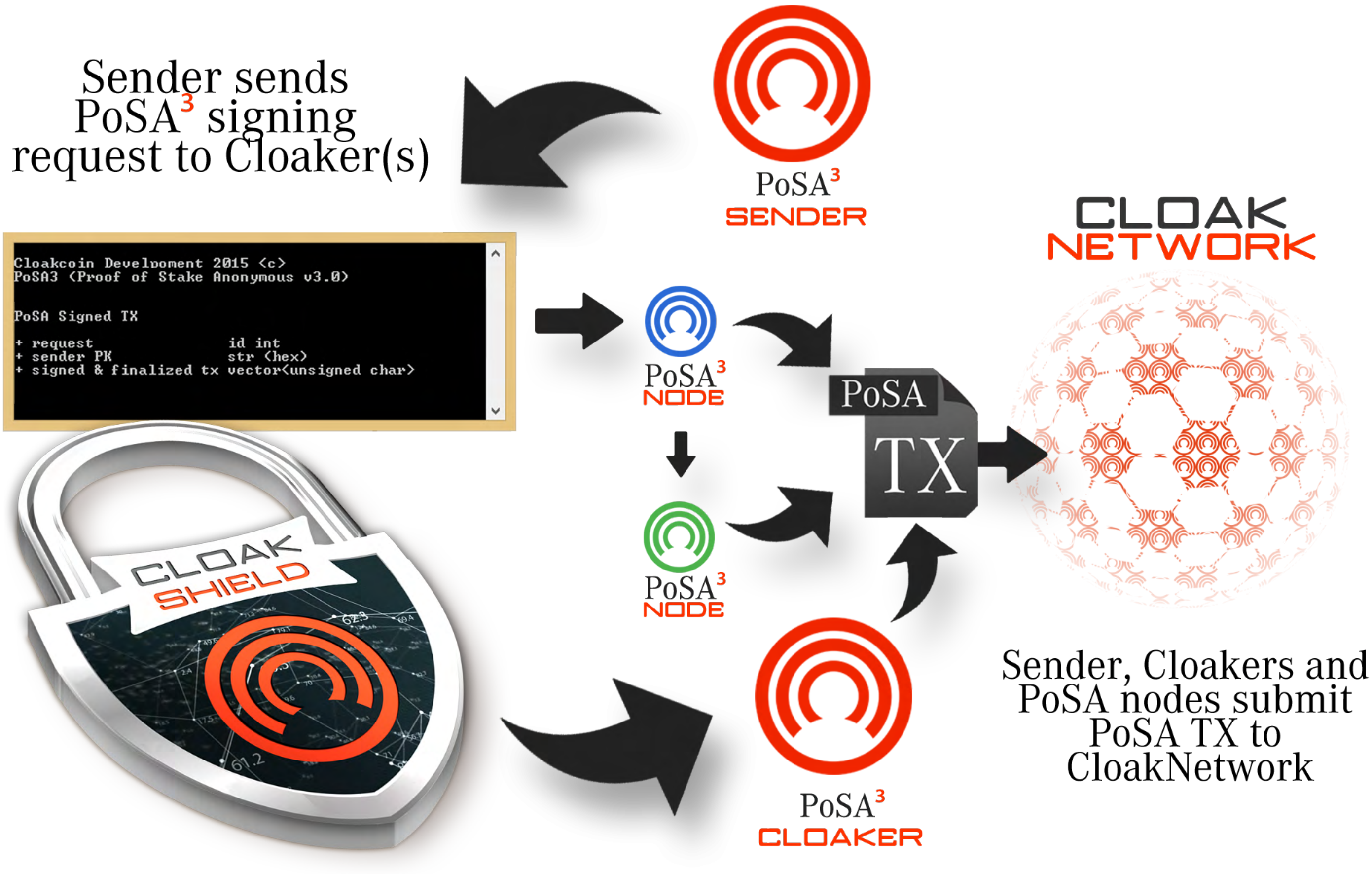
```
Cloakcoin Develpoment 2015 (c)
PoSA3 (Proof of Stake Anonymous v3.0)

PoSA Signing Request

+ request id int
+ sender PK   str (hex)
+ unsigned tx vector<unsigned char>
```

```
CloakCoin Development 2015 (c)
PoSA3 (Proof of Stake Anonymous v3.0)

PoSA Sign Response

+ request id  int
+ cloaker PK  str (hex)
+ signed tx vector<unsigned char>
```

Sender sends PoSA^3 signing request to Cloaker(s)

Cloaker(s) sign the PoSA^3 transaction and send encrypted TX back to Sender

Step 3 *- Sender Signs PoSA TX, relays TX to Cloakers and submits TX to CloakCoin network*



Sender sends PoSA³ signing request to Cloaker(s)

```
Cloakcoin Develpoment 2015 <c>
PoSA3 <Proof of Stake Anonymous v3.0>

PoSA Signed TX

+ request                  id int
+ sender PK                str <hex>
+ signed & finalized tx vector<unsigned char>
```

Sender, Cloakers and PoSA nodes submit PoSA TX to CloakNetwork

5. The Future of PoSA – Further Development

PoSA forms the core of CloakCoin and will continue to be developed and improved as we move forward with OneMarket. Here are some of the features planned for 3.x revisions:

Improved Proof-of-Stake Algorithm

Proof of Stake (PoS) is a method of securing a cryptocurrency network that relies upon users showing ownership of coins in order to sign blocks. In the long run, the probability of signing blocks is proportional to the amount of coins owned, someone owning 1% of total coin supply will be able to sign 1% of all proof of stake blocks. Compared to proof of work approach, proof of stake requires significantly less computational power, and thus less energy usage.

Coin Age and Linear Proof-of-Stake

Fundamental to most implementations of Proof of Stake, including that of CloakCoin, is the concept of Coin Age. Essentially, this is a measure of how long a coin holder has held onto coins without spending or moving them. From the time a transaction is completed, coins that were part of that transaction begin to accumulate Coin Age (which starts at zero). In its simplest form, entitled "linear coin age", coins will accumulate a minute/hour/day/year of Coin Age each minute/hour/day/year of age. For example, a person that holds 365 coins for 100 days accumulates 36,500 'coin days', or approximately 100 'coin years' (A 'coin year' is defined to account for leap years, and thus is not exactly 365 days, but ~365.24 days).

Linear Proof-of-Stake designs have attracted criticism in relation to Coin Age. Many argue that linear Proof-of-Stake encourages hoarding of coins (which can have a detrimental effect on trade and transfer volume). Another valid complaint against linear Proof-of-Stake relates to the effect it can have on network security. Linear Proof-of-Stake implementations often suffer due to users periodically connecting to the Cloak network to stake their coins and then disconnecting once all Coin Age has been destroyed. The user then waits until Coin Age has replenished before repeating the connect-stake-disconnect process. This does not provide the best security for the network, and a Proof-of-Stake algorithm that rewards frequent or constant`r staking would be most beneficial to CloakCoin and related Proof-of-Stake currencies.

To ensure PoSA Cloakers are rewarded as amply as possible, Coin Age should be removed from CloakCoin's Proof-of-Stake algorithm. This would ensure that Cloakers receive both the full staking reward and any PoSA Cloaking rewards. The additional incorporation of a velocity component in calculating staking rewards would further reward active PoSA Cloaking nodes, encouraging users to participate in PoSA Cloaking to further increase their earned interest in addition to earned Cloaking rewards. In addition to providing greater rewards to actively participating users, an improved Proof-of-Stake algorithm also provides the aforementioned improvements to network security.

Combining and Splitting PoSA Transactions

PoSA 3.0 currently creates a single 'Cloaked' transaction per transfer.
We are currently working on an update to the PoSA framework that will allow multiple PoSA transactions to be combined into a PoSA super-transaction. This will effectively contain multiple 'Cloaked' transactions and provide even greater anonymity for Cloak users. This extension will allow users to select the number of co-operative PoSA transactions they require in addition to the number of Cloakers.
This addition of course remains fully decentralized and trustless.

Another PoSA sending enhancement currently being fleshed-out by the Cloak Team is the ability to 'Cloak' a large amount of Cloak as a series of smaller PoSA transactions.
To achieve this, a user would choose the amount of Cloak they would like to send Cloaked to an address. CloakCoin would then work in the background to create a number of smaller PoSA transactions of an even amount, which can be Cloaked and submitted to the Cloak network over a set period of time. This batching process will be compatible with 'combined' PoSA transactions, providing further Cloaking protection for transfers.

Deterministic Wallet Addresses

PoSA3 often utilizes change addresses as part of the Cloaking process. The addition of a hierarchical deterministic wallet (see BIP 0032) will allow a wallet to be recreated using the same 12 word mnemonic of common English words. This will preserve change address keys without the need to backup the wallet itself and also allow PoSA session keys and OneMarket keys to be secretly linked by the owner.

6. FAQ

Q. How do Cloakers assist a PoSA transaction?

A. Cloakers provide one or more inputs that are used to ‘Cloak’ the input from the sender. Cloakers also supply a series of return addresses which return their input and also reward the Cloaker with a fee. The return addresses are chosen carefully in order to prioritize addresses with activity. This makes it much harder for anyone performing blockchain analysis to pinpoint the true output of a PoSA transaction. The PoSA system will also check the target address so that ‘cloaked’ outputs mirror the true output as closely as possible.

Q. I heard PoSA v1 had problems with trust. How is that addressed with PoSA v3.0?

A. PoSA v1 used a series of ‘hops’, with each ‘hop’ splitting the amount and forwarding the parts to the next PoSA node. This raised concerns as one or more ‘hop’ nodes could retain the funds as opposed to passing them on. PoSA 3.0 combines all inputs and outputs into a single transaction to ensure all parties in the transaction are funded simultaneously or not at all.

PoSA v3.0 combines all inputs and outputs into a single transaction to ensure all parties in the transaction are funded simultaneously or not at all.

Q. How long do PoSA transactions take to complete?

A. PoSA transactions are currently allotted one minute to complete. Cloaking nodes helping to ‘Cloak’ a PoSA transaction will reserve the necessary funds until the PoSA transaction completes or the allotted time expires. In the case of an expired or aborted PoSA transaction, funds are unlocked locally for re-use.

Q. How does PoSA affect staking?

A. Any coins used in a PoSA transaction (as a Sender or Cloaker) will have their coin-age reset. It should be noted however, that participating in Cloaking should provide a much higher return than staking. The Cloak Team are working to revise the PoSA algorithm for the upcoming hard-fork release (PoSA 3.1). Please see Section 5 - 'The Future of PoSA – Further Development' for more details.

Q. Do I need a certain amount of Cloaks in my wallet balance to be a PoSA Cloaker?

A. You can offer your services for Cloaking regardless of the balance in your CloakCoin wallet. When PoSA Cloaking is enabled, CloakCoin will reserve a portion of your balance for participating in PoSA Cloaking, for which you will earn a Cloaking reward. The default reserve amount is ~50%, but this value can be adjusted by the user. The chosen value with be randomized in order to prevent linking of PoSA announcements by advertised Cloaking balance.

It should be noted that wallets with a higher balance have a higher chance of being chosen as a Cloaker as they are more likely to have the required Cloaking balance available for larger PoSA transactions.

Q. How does this protect against a time based attack where someone looks on the blockchain for identical inputs and outputs?

Would it not be possible to discern the destination this way?

A. PoSA transactions group the outputs and are ensured to have multiple matching output amounts to 'cloak' the recipients output.

Q. Can the originator of a PoSA transaction be determined by examining the script signature to determine signing order?

A. No. During the signing process, script signature order is randomized when combining the signatures. This is done by the sender and the participating Cloakers.

Q. Can an eavesdropper monitor the network to watch for outgoing PoSA transactions being submitted to the network to determine the true sender?

A. No. A PoSA transaction is submitted to the network by all parties in a random order.

This provides mitigation against such eavesdropping attacks.

Q. What is the fee for a PoSA transaction?

A. The PoSA transaction fee is 1.8% of the sent amount, plus network fees. This is used to reward PoSA nodes that assist with cloaking a PoSA transaction.

Q. Does PoSA 3.0 require a hard-fork of the Cloak network?

A. No. Older CloakCoin clients will handle PoSA transactions without issues, but they will not be able to create them or participate in 'cloaking' them. The next revision of PoSA3.x however, will require a hard-fork due to changes to the underlying Proof-of-Stake algorithm, and support for additional script opcodes for OneMarket features (such as Block Escrow).

Q. How does PoSA 3.0 protect against 'bad actors'?

A. The PoSA system features extensive DDoS protection to 'blacklist' nodes for the duration of a session. If a PoSA node repeatedly refuses to sign, they will be excluded from PoSA Cloaking invitations for the remainder of the current session. We are currently researching additional methodologies for further penalizing uncooperative PoSA nodes and will likely implement a system that requires Cloakers to escrow a nominal, refundable fee that could be claimed as a penalty in instances where a node attempts to block a PoSA transaction by refusing to sign the finalized transaction. It should be noted that whilst malicious nodes may attempt to hamper a PoSA transaction, they are not able to steal or misappropriate any funds.

Q. What is the maximum number of Cloakers that can assist in a PoSA transaction?

A. The maximum number of Cloakers is fixed at 25. The PoSA system is flexible and this number can easily be extended.

5. References

[01] http://bitcoin.org

[02] https://en.bitcoin.it/wiki/Category:Mixing Services

[03] https://wiki.openssl.org/index.php/Elliptic_Curve_Diffie_Hellman

[04] http://blog.ezyang.com/2012/07/secure-multiparty-bitcoin-anonymization

[05] http://blog.atmel.com/2014/10/21/ecdh-key-exchange-is-practical-magic

[06] https://bitcointalk.org/index.php?topic=279249.0 (CoinJoin: Bitcoin privacy for the real world)

[07] https://bitcointalk.org/index.php?topic=27787.0 (Proof of stake instead of proof of work)

[08] https://en.bitcoin.it/wiki/Proof_of_Stake

[09] https://en.bitcoin.it/wiki/Deterministic_wallet

[10] https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

[11] http://www.onion-router.net