OUSD: Private, Instant, USD

OUSD is a decentralised DAI backed, Cryptonote coloured coin, issued on the Oxen Blockchain

Fenwick Feniks
CryptoNote Developer
056574129abd8a81ea6454fef608d10a
658e6c56ec5b32915a605b4b07fb6a2f3c

Borislav Babochka
CryptoNote Developer
05f7c71c89e0f0453466b12bc1e2bd74
12e49133b9c099ac5b7a94dfd0be913f52

***Abstract*—The popularity of stablecoins is increasing massively. The collective market capitalization of stablecoins has already grown to 80 billion USD as individuals, businesses and traders seek both insulation from the high volatility of cryptocurrency markets as well as the benefits of transacting on a decentralised ledger. However, existing stablecoins have significant privacy shortfalls. Most are built on entirely transparent blockchains, allowing for the origin, destination and amounts of transactions to be traced by anyone. OUSD is a new stable, coloured coin on the Oxen blockchain, providing full CryptoNote privacy, instant settlement, low fees, and 1:1 backing in DAI.**

I. How It Works

OUSD is a very simple construct, with the least moving parts possible. This is an intentional design decision to make OUSD extremely easy to mint, redeem and audit. Its stability is derived from DAI, one of the oldest, most trusted, decentralised stablecoins. DAI exists as an ERC20 token on the Ethereum network.

To better understand the protocol, it is important to describe the most important parts of the protocol minting OSUD, redeeming OUSD and auditing the supply of OUSD.

Minting new OUSD

Users mint new OUSD by calling the DepositWithAddress function in the OUSD Ethereum smart contract. This function accepts a DAI deposit along with an associated Oxen address. Once a user sends DAI to the OUSD contract, the only way to reclaim the DAI is by calling the claim function, which requires a number of conditions be met (See Redeeming OUSD).

After DAI has been deposited into the OUSD contract, it is necessary for the service node network to witness this deposit on the Ethereum network and collectively agree on its validity. To be able to do this, each service node will need to have access to a queryable Ethereum API. This can be achieved by either integrating with an existing provider Ethereum API provider like Infura into the service node or allowing a service node to run a light Ethereum node.

Every 1100 Ethereum blocks (approximately 4 hours), the service node network will come to consensus about incoming transactions to the OUSD contract address. To reduce network gossip, each service node produces a signed coinbase transaction which mints OUSD to the Oxen addresses which are deterministically selected by the witnessed transactions on the Ethereum chain. This Coinbase transaction is gossiped into the Oxen mempool by each service node. Service nodes then additively combine the EdDSA Signatures into a single 64 byte threshold Schnorr signature, once a threshold of 85% is reached the network then accepts that Coinbase transaction as valid and it is included in the next block by the next service node block production quorum.

Redeeming OUSD

To redeem OUSD back into DAI users will first send their OUSD to a Oxen burn address, This burn address is chosen such that the the public view key and public spend key are known, however no public spend key is known. This allows all service nodes to observe new incoming transactions to the burn address. When sending an amount to the burn wallet, the user will also specify in the TxExtra field of the transaction which Ethereum address to deposit the DAI into.

Service nodes will witness each transaction to the Oxen burn address. When witnessing a transaction, each node will produce an EdDSA signature from their Service Node ED25519 key. This EdDSA signature sign data containing the transaction hash, amount to be minted in DAI, and the Ethereum address to mint to. They share this signature with other service nodes. Once a predetermined threshold of signatures has been shared, service nodes can use Schnorr signature aggregation to generate a single 64 byte proof that indicates 85% of the service node network has come to consensus on an Oxen transaction.

Proving a burn transaction on the Oxen network entitles the OUSD burner to an equal amount of DAI, which can be withdrawn from the OUSD Ethereum smart contract. Validation of burning on the Oxen blockchain inside the OUSD Ethereum contract is possible because the OUSD contract keeps track of the current aggregate public key (XXX BYTES) for the service node network. The OUSD contract has an updateKey function which can be called by any user if they can provide a signature from a threshold of 85% of the previous aggregate key on a new aggregate key. Using the current aggregate key, the OUSD contract can validate threshold Schnorr signatures from the service node network inside the contract using EVM compatible signature verification contracts developed by Chainlink [1].

Thus, any user can request this transaction burn proof from any service node. This proof is then taken to the Ethereum blockchain alongside the signed data, and a transaction is submitted to the OUSD smart contract under the withdrawal function. The proof is validated inside the contract and, if valid, the contract allows withdrawal of DAI to the specified Ethereum address.

Auditability

The supply of OUSD is easily auditable by any individual, since the amount of OUSD minted is the same as the DAI which exists in the Ethereum OUSD vault minus any DAI deposits yet to be witnessed and converted into OUSD. OUSD on the Oxen blockchain is transparently minted through coinbase transactions allowing the total supply of OUSD on the Oxen blockchain to be fully audited. If the total OUSD supply matches the total DAI supply in the OUSD Ethereum contract minus OUSD to DAI conversions yet to be claimed, then the supply is balanced.

II. Blockchain Specifics

OUSD token

OUSD is a coloured coin on the Oxen blockchain, the construction is based on the MARCT construction detailed in the CLSAG paper [2]. This allows for the construction of colored coins using the d-CLSAG signature type by adding a specific commitment to that asset type, in this case the OUSD colored coin.

This allows OUSD to maintain the same privacy properties as Oxen, having the protection of ring signatures, stealth addresses, and range proofs which obfuscate the sender address, recipient address, and amount of a transaction. Minting OUSD occurs through coinbase transactions from the service node block production quourms.

Blink

Since OUSD is issued as a token on the Oxen blockchain, it can inherit Blink [3] , which will allow OUSD to be sent and instantly confirmed on the Oxen blockchain. This allows merchants to instantly accept OUSD as payment without waiting for any prerequisite number of blocks to confirm payment and provide goods or services to a customer.

III. Economics

Value Accrual for Oxen

The creation and maintenance of OUSD will require additional work for the Oxen network, including: witnessing transactions on the Ethereum chain and coordinating to provide signatures for withdrawal requests from the Oxen blockchain. Because of this, there should be some sort of value actual mechanism for Oxen baked into OUSD.

To achieve this, we can stake all deposited DAI into decentralised lending protocols or as liquidity in automated market makers. Projects like Curve, YFI, and Compound finance are obvious candidates. Currently, these projects typically produce a yield of 6-12% per year. This yield can be captured in the contract and used to buy and burn wOxen (the ERC20 wrapped version of OXEN) in the wOxen-USDT Uniswap pool. This wOxen and corresponding Oxen would then be burned, lowering the circulating supply of Oxen and leading to a price increase for the existing tokens.

Minting and Burning Fees

Minting new OUSD is free because value is accrued through the staking of user deposited DAI into various yield producing protocols.

Redeeming OUSD to DAI does incur a fee of 10 OXEN. This fee is levied to discourage low volume redemption claims which require significant network communication and computation. Instead the preference is that users sell OUSD on the open markets where market makers buy OUSD at a slight discount, market makers can then bundle orders together and convert them back into DAI at a slight profit. The 10 Oxen fee is burned to further increase network sustainability.

IV. Attacks & stability

there are three cases we want to ensure the OUSD smart contract can recover from if encountered: removal of inactive nodes, dishonest minorities, and dishonest majorities.

Inactive Nodes

Because OUSD can only be redeemed with the approval of a 85% of the current service nodes, we need to be swift in updating the aggregate key in the Ethereum smart contract to ensure the contract always has an aggregate key which closely reflects the current state of the network. If the key becomes out of date, it's possible the network could reach a state where it was unable to sign any valid statement from the currently supported aggregate key, meaning DAI would be stuck in the OUSD contract.

Anyone can call the updateKeys function in the OUSD contract if they provide a new aggregate key which is signed by a threshold of the old aggregate key. This process should only need to be completed every few weeks as the service node network grows and contracts, However, calling this function does incur a 45,000 GAS fee, which could discourage average users from regularly calling the function. It is expected the OPTF will initially pay the GAS to perform this duty and ensure the key in the contract is kept up to date.

In the future, the contract could refund this GAS cost plus a small incentive payment by siphoning a portion of the returns earned from DAI yield farming. This would encourage profit seeking parties to regularly call the updateKeys function.

Dishonest Minorities

Inactive nodes act differently to actively dishonest minorities, since inactive service nodes do not vote on the validity of witness transactions from the Ethereum or Oxen blockchain. Dishonest minorities actively vote against the witness transactions which are valid blockchain transactions. They may do this hoping to freeze the minting and redemption of OUSD. Such an attack would require control of 16% of the service

node network, and could be used to extract a ransom from users of OUSD who want to be able to recover their DAI from the Ethereum smart contract.

To prevent this, we can add a governance recovery mechanism in the OUSD Ethereum smart contract, this mechanism can only be triggered if no withdrawals have been made from the OUSD Ethereum smart contract in 1 week, which ensures that in normal operation this function cannot be called.

In the event the OUSD contract has had no interaction in the last two weeks, it would be clear there is a network issue and the DAI in the OUSD contract needs to be recovered to the relevant holders of OUSD. Non-signers to the OUSD withdrawal proofs would be deregistered as service nodes and the governance key would update the aggregate key in the OUSD smart contract. From here, the network can continue operation with the superminority of dishonest service nodes removed from the network.

The exact parties who hold control of the governance key should be decided by the community, but I would suggest this functionality be held initially by the OPTF.

Dishonest Majorities

Dishonest majorities are groups of 86% or more of the service node network who act against the rules of the protocol, typically the goal of a dishonest majority is to carry out economically advantageous attacks, like stealing DAI from the OUSD contract.

We can consider OSUD economically safe from these attacks if the value that would be derived from the attack is less than the value that would be lost if the attackers were identified and had their staked Oxen slashed. For example the current market cap of Oxen is $107 million USD, and about 50% of the Oxen supply (about $53.5 million USD) is already locked into the service node network. 86% of that is $46 Million, meaning that any attack which stole less than $46 million DAI would be unprofitable.

With no other precautions in place this sets a hard cap on the supply of OSUD which can safely exist at 86% of the value of the current OXEN staked on the network. This should provide enough capacity initially, and as the market capitalization of Oxen increases, so too will the economically safe amount of circulating OUSD.

There are other factors alongside natural growth of the Oxen market capitalization we may wish to consider to further increase the amount of OUSD which can safely be minted. One possible route would be to set a maximum amount of DAI, say 10% of the total DAI in the contract, which could be withdrawn from the OUSD smart contract in a 24 hour timeframe. This limit would be enforced on the Ethereum network through the OUSD contract.

Since dishonest majorities are easily detectable by honest actors, this would limit the amount a dishonest majority could steal to only 10% of the value of the DAI contract. Once the theft had occurred, the dishonest majority would be detected and a hardfork could be initiated which burns the dishonest service nodes Oxen, the governance key could then be used to rotate back to an honest service node set. This change would allow the economic maximum of OUSD minted at the current Oxen market capitalisation to be $460 million.

Detection Techniques

There are various scenarios in which we need to slash or remove service nodes for either signing dishonest witness transactions or refusing to sign honest burn proofs. However, this requires detection of exactly who signs a proof or transaction, which is not possible with general Schnorr signatures constructions.

It is possible, however, to detect signing/non signing at the p2p layer, before aggregation, however collecting this information at a global network scale and allowing for the audit of that information creates scalability issues. To create accurate logs, I suspect it will be easier to lean on existing service node swarms to create a two tiered signature aggregation process. This system should mean service nodes only need to keep track of nodes in their own swarm while still providing useful information if an audit is conducted.

Each swarm would have an aggregate key composed of the 5-10 service nodes public keys in their swarm. When a new transaction proof or witness transaction needs to be signed, the service nodes would sign the data locally and share the signature with each node in their swarm. Each service node also saves the signatures they receive from the other nodes in their swarm during the round. Each node in the swarm would then aggregate the signatures it received and (if it reached the threshold required for the swarm) submit the signature to the network. Those signatures from individual swarms are then aggregated further and submitted to the OUSD Ethereum contract when proofs are required.

If a dishonest majority signed a transaction stealing DAI or a dishonest minority blocks the network from signing, the network can begin an audit. In an audit each node submits all signatures gathered from its swarm in the last round of signing. These logs can be combined to identify which nodes are signing dishonest transactions or refusing to sign honest transactions.

V. Threat Model

The OUSD threat model is extremely simple to evaluate for users of OUSD, and consists of the following assumptions

- Deposits to the OUSD Ethereum smart contract cannot be reversed after a reasonable number of Ethereum block confirmations
- Deposits to the OUSD burn address cannot be reversed after a being checkpointed
- No more than 85% of the Service Node network is dishonest
- DAI maintains its peg to the US dollar

VI. Additional Considerations

Frozen Coins

Centralised stablecoins like Tether and USDC allow the ERC-20 smart contract operator to freeze coins for any reason that the operator sees as an illegitimate use of that stablecoin. If we

allowed deposits to the smart contract of USDC and USDT, then the system runs the risk that admin keys could be used to lock coins deposited to the smart contract. The risk profile is increased by the fact that a fully private stablecoin cannot be traced. Such interventions would cause coins locked in that contract to become worthless, this would in turn impact the peg of OUSD. To prevent this, the OUSD minting smart contract only allows deposits of DAI, an asset which does not support freezing.

Non USD Assets

The described scheme can work to securely peg any ERC-20 asset into the Oxen blockchain as a coloured coin. This means Euros (EURS), Ethereum (wETH), Bitcoin (wBTC,renBTC) and even Gold (PAXG, PMGT) can all be represented as a private coloured coin on the Oxen blockchain.

VII. Conclusion

Oxen is uniquely well positioned to build a private stablecoin, it already has a large established service node network which can trustlessly witness events external to the Oxen blockchain. It already has an established private CryptoNote blockchain built from the foundations of Monero, and it already has an instant settlement system called Blink. The combination of these features allow for OUSD, an instantly settled, fully private and 1-1 DAI backed stablecoin.

References

[1] *EVM compatible signature verification contracts*, https://blog.chain.link/threshold-signatures-in-chainlink/.

[2] *Compact Linkable Ring Signatures and Applications*, https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2019/654&version=20190924:130827&file=654.pdf.

[3] *Blink improvement proposal*, https://github.com/oxen-io/oxen-improvement-proposals/blob/master/LIPS/LIP-4.md.