CLOAG

ENIGMA V2.1

ENIGMA

A PRIVATE, SECURE AND UNTRACEABLE TRANSACTION SYSTEM FOR CLOAGCOIN

1. ABSTRACT

CLOAG Coin is a cryptocurrency designed to facilitate private, secure and untraceable decentralized transfers with Enigma.

CLOAG is a dual PoW/PoS (Proof of Work, Proof of Stake) coin, which is now in the Proof-of-Stake (interest bearing) stage.

Enigma is CLOAG Coin's private, secure and untraceable payment system, that forms the basis of future development and provides the underlying transaction system for the decentralized applications running on the CLOAGCoin network.

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.

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

2. ENIGMA V1.0 OVERVIEW

Enigma is the first public iteration of CLOAG's private, secure and untraceable payment system. Enigma transactions are 'CLOAGed' by other users, who receive a reward for their assistance. The other users provide

inputs and outputs to the Enigma transaction making it impossible to determine the true source and destination of the CLOAG transfer. All Enigma messages on the network are hashed and encrypted for the recipient using

CLOAG Shield to ensure data security and integrity. Please see Section 3 – 'CLOAGShield' for more details.

2.1. THE ENIGMA PROCESS (FOR ENIGMA ENABLED NODES)

ENIGMA ANNOUNCEMENTS

Enigma nodes communicate over the CLOAG network and a node will keep track of other active Enigma nodes. Enigma Announcement Broadcasts

alert other Enigma nodes of our public session key and current Enigma CLOAGing balance.

ENIGMA CLOAGING REQUESTS

When a user wishes to send a CLOAGed Enigma transaction, they elect a series of Enigma nodes (with a high enough Enigma balance) and request their assistance in CLOAGing. An Enigma node can choose to assist in CLOAGing and send an acceptance response to the requester to indicate this. If an Enigma node declines to participate in CLOAGing or does not respond in a timely manner, an alternate Enigma 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 an Enigma transaction or refuses to relay Enigma messages. Enigma CLOAGing nodes use an Elliptic Curve Diffie Hellman key exchange (ECDH) to derive a shared secret with the Enigma initiating node, which is used to generate a shared secret key for symmetric RSA-256 data encryption between a CLOAGing node and the sender node.

ENIGMA CLOAGING ACCEPTANCE

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

THE 'CLOAGED' ENIGMA TRANSACTION

The Enigma Sender constructs a 'CLOAGed' transaction using the inputs and outputs provided by the Enigma CLOAGer nodes. The Enigma Sender then adds their own inputs and outputs to the transaction, before shuffling all transaction inputs and outputs to facilitate 'CLOAGing'. The 'CLOAGed' transaction is then encrypted and sent (using CLOAGShield) to each participating CLOAGer. CLOAGer nodes check the transaction to ensure the inputs and outputs they supplied are present in the 'CLOAGed' 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 Enigma Sender. Once all Enigma CLOAGers have signed the transaction, the Enigma Sender confirms the signed transaction is valid and signs it. The ‘ CLOAGed’ transaction is then ready for submission to the network.

2.2.1. TRACKING ENIGMA CLOAGING NODES

Enigma enabled nodes on the CLOAG network broadcast announcements to other nodes. These Enigma announcements contain the public ec-key ID of the node and the currently available balance for Enigma CLOAGing operations. Nodes maintain a list of other active Enigma nodes on the network so that they can communicate for CLOAGing 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 CLOAGing balance for the session periodically to other nodes on the CLOAG network.

3. Nodes keep track of other active Enigma CLOAGing nodes and can communicate with them directly or indirectly (via CLOAGShield Onion Routing).

2.2.2. INITIATING AN ENIGMA TRANSACTION

ALICE wishes to send 10 CLOAG to BOB using 5 mixer nodes.

1. Alice broadcasts an Enigma request to the network, containing her public Enigma session key and the amount of CLOAG she wishes to send. Her request is securely routed through a series of 5 Enigma nodes to mask the originator.

2. Catherine has 'CLOAGing Mode' enabled and creates a secure CLOAGShield encryption channel for secure communication with Alice. Catherine then constructs an Enigma response packet and sends it securely to Alice. The response contains a list of Catherine's inputs and outputs that Alice will use to 'CLOAG' her transaction.

3. Alice decrypts and processes Catherine's Enigma response and creates an Enigma transaction using her own inputs and outputs mixed with Catherine's inputs and outputs. This is encrypted and sent to Catherine for signing.

4. Catherine decrypts the Enigma 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 Enigma transaction passes the tests, Catherine 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 submitted to the network (securely routed through Enigma nodes) for inclusion in a block.

6. When the transaction is finalized, Bob will receive the funds from Alice and Catherine will receive a 'CLOAGing' reward for assisting in the Enigma transaction.

7. Due to Catherine's inputs and outputs mirroring Alice's, it is not possible to ascertain the true sender and recipient of the Enigma transaction.

ENIGMA TRANSACTION EXAMPLE

ALICE wants to send coins anonymously to BOB.

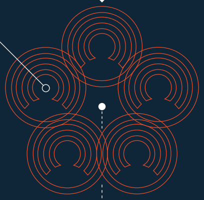
ALICE (−10.0992) CLOAG

(−10) CLOAG + (−0.0992) Enigma fee
= (−10.0992) CLOAG total

ENIGMA mixer nodes begin communicating.

CATHERINE

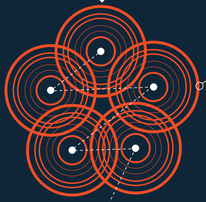
Every coin holder can announce themselves as a Mixer Node, also known as a "CLOAGer".



Every participant remains anonymous and communicates through an encrypted channel.

ALICE's wallet is now connected to mixer nodes.

Each mixer node helps ALICE by shuffling around the transaction.



This network of nodes creates decentralized anonymization similar to TOR Onion Routing.

Mixer nodes get rewarded for CLOAGing ALICE's transaction.

(+0.0992) CLOAG

A linear fee from .2% (>1000 coins) to 1% (0 coins) is shared amoungst all participating CLOAGers.

The system works seamlessly to ensure complete anonymity and total privacy.

BOB then receives ALICE's encrypted payment

BOB (+10) CLOAG

BOB successfully receives 10 CLOAG anonymously.

3. CLOAGSHIELD

CLOAGShield provides secure communications between nodes on the CLOAG 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). CLOAGShield is designed to secure both Enigma and decentralized CLOAGCoin applications, and will ensure your data stays as private as possible.

CLOAGShield 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. CLOAG Shield uses ECDH to accomplish this:

- Alice has Enigma private key dA and Enigma public key QA=dAG (where G is the generator for the elliptic-curve). Bob has Enigma private key dB and Enigma public key QB=dBG.

- Alice has Bob's Enigma public key dB from the Enigma announcements he sends to the network to announce his availability for CLOAGing 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 OpenSSLEVP_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 CLOAGShield secured messages for Bob.

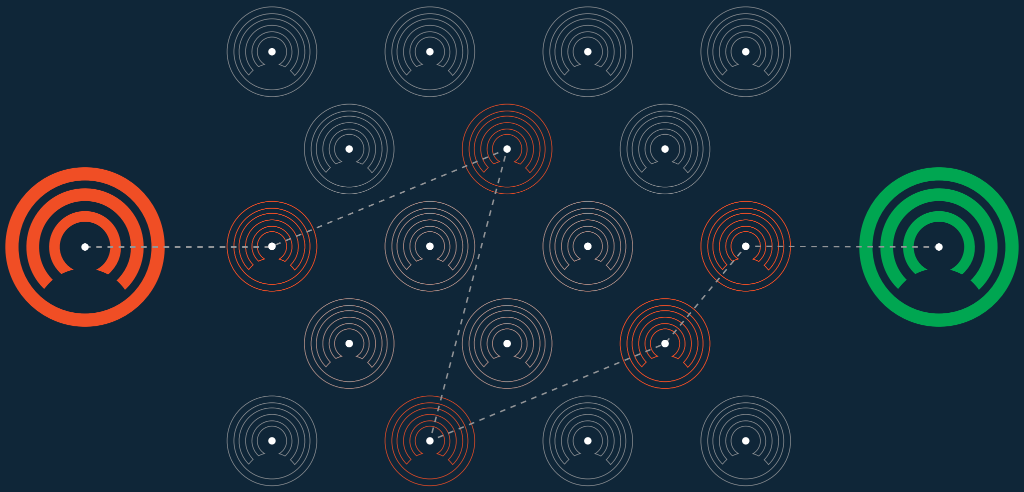
When Bob receives a CLOAG Shielded message from Alice, he reads Alice's public key from the message header and generates the same shared secret key as Alice, as per the steps above (with his secret key, instead of Alice's).

The CLOAG wallet maintains a list of active CLOAG Shield keys and will check the list for an existing CLOAGShield key before generating one.

CLOAGSHIELD DATA

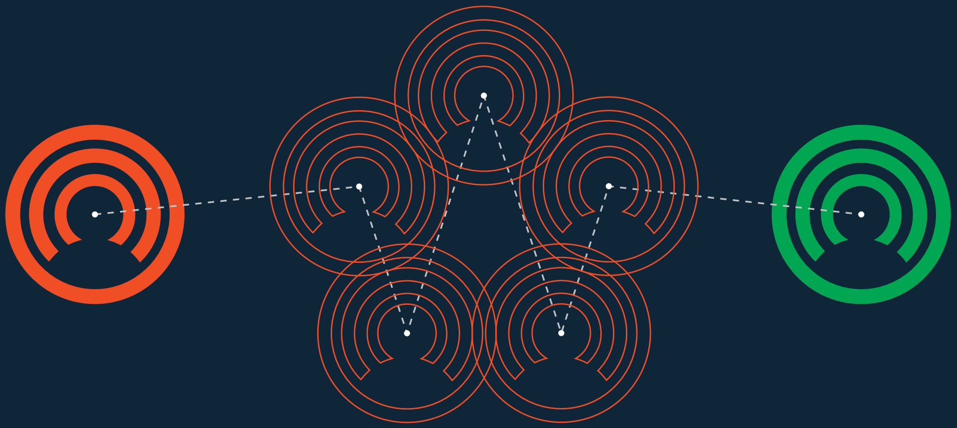
CLOAGShield allows any CLOAG data objects to be serialized and transmitted securely to one or more recipients. A CLOAGShield data packet-header contains the sender's Enigma public key and the public keys hashes of the recipients.

CLOAGShield 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 CLOAGShield data to ensure that the recipient info in the header matches the encryption key, and that the data has not been altered.



CLOAGSHIELD 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 Enigma network (using CLOAG Shield) 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 CLOAGCoin network. When an Enigma node wishes to communicate with another Enigma node it selects a number of other Enigma 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 Enigma network, exit nodes are not required and CLOAGShield ensures there is no risk of a relay node reading or altering the encrypted data.

4. STEALTH ADDRESSES

CLOAG uses the Enigma system to faclilate private/secure transactions.

CLOAGSHIELD - NODE TO NODE COMMUNICATIONS

On startup, each CLOAG wallet generates a [NID_secp256k1] keypair (CLOAGing Encryption Key / CEK) to enable them to derive ad-hoc secrets using ECDH with their private key and the recipient's public key. This

communication forms the basis on all node-to-node communications relating to Enigma. See 'src/enigma/CLOAGshield.h/.cpp' for more information on this. This ECDH based encrypted communication is also utilized for onion-routed data, which is handled by CLOAGShield. When onion routing is enabled, the client will attempt to construct a valid onion route for the data using the list of Enigma peers that it is aware of. The node may not have a direct connection to the Enigma peers, but that is not necessary as CLOAGData (data packed for routing with CLOAGShield) packets are relayed peer-to-peer. An onion route will typically consist of 3 distinct routes to the destination node, with 3 node hops per route. Multiple routes are used to cope with situations where a routing node drops offline.

Nodes periodically send out an Enigma Announcement (src/enigma/enigmaann.h) to peers to advertise their services for onion routing. Other nodes on the network store the announcements (until they expire or are replaced with an update) and use them to construct the onion routes.

STEALTH ADDRESS TRANSACTION EXAMPLE

When a node sends an Enigma transaction to a stealth address, the following happens:

1. Sender generates inputs to cover amount sent, Enigma reward and network fee (1% at 0 coins thru .2% at 1000 and higher coins).
2. Sender generates a CLOAGingRequest object (containing unique stealth nonce for this request).
3. Sender generates between 2 to 4 one-time stealth payment addresses using the recipients stealth address and splits the sent amount randomly between the addresses.
4. Sender decides how many participants are going to be used. From 5-25 participants can be chosen (each participant gets 80-120% of an equally split Enigma fee).
5. Sender onion routes CLOAGRequest to network. The request contains the 'send amount' so that CLOAGers know how much to reserve.
6. CLOAGer picks up CLOAGRequest and decides to participate.
7. CLOAGer supplies X inputs to sender and a stealth address and stealth hash (for their change).
8. CLOAGer sends CLOAGingAcceptResponse to Sender. This contains stealth address, stealth nonce and TX inputs.
9. Sender waits until enough CLOAGers have accepted.
10. Sender creates Enigma transaction using own inputs and CLOAGer inputs. Inputs are shuffled.
11. Sender creates TX ouputs for all CLOAGers. The outputs randomly split their change and return it to them. This also allocates the CLOAGing reward to CLOAGers.

12. Sender creates their own change returns for the Enigma TX. These are one-time stealth payment addresses.

13. The Sender calculates the network TX fee and subtracts this from their own change return.

14. The Sender sends the Enigma TX to the CLOAGers for signing.

15. CLOAGers check the TX to ensure their inputs are present and correct and that there are one-time payment addresses linked to one of thier stealth addresses with payment that exceeds the input amount.

16. CLOAGers sign or reject the TX and send signatures to Sender.

17. Sender collates the signatures and transmits the finalized, signed TX to the network.

18. Nodes scan incoming transactions for stealth payments and Enigma payments and detect any payments or change. Keypairs and addresses are generated for any matching payments and generated keys/addresses are saved to the local wallet.

5. THE FUTURE OF ENIGMA – FURTHER DEVELOPMENT

Enigma forms the core of CLOAGCoin and will continue to be developed and improved as we move forward with CLOAGCoin. Here are some of the features planned for future 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 CLOAGCoin, 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 CLOAG 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 staking would be most beneficial to CLOAGCoin and related Proof-of-Stake currencies.

To ensure Enigma CLOAGers are rewarded as amply as possible, Coin Age should be removed from CLOAGCoin's Proof-of-Stake algorithm. This would ensure that CLOAGers receive both the full staking reward and any Enigma CLOAGing rewards. The additional incorporation of a velocity component in calculating staking rewards would further reward active Enigma CLOAGing nodes, encouraging users to participate in Enigma CLOAGing to further increase their earned interest in addition to earned CLOAGing 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 ENIGMA TRANSACTIONS

Enigma currently creates a single 'CLOAGed' transaction per transfer.

We are currently working on an update to the Enigma framework that will allow multiple Enigma transactions to be combined into a Enigma super-transaction. This will effectively contain multiple 'CLOAGed' transactions and provide even greater anonymity for CLOAG users. This extension will allow users to select the number of co-operative Enigma transactions they require in addition to the number of CLOAGers.

This addition of course remains fully decentralized, private and secure. Another Enigma sending enhancement currently being fleshed-out by the CLOAG Team is the ability to 'CLOAG' a large amount of CLOAG as a series of smaller Enigma transactions. To achieve this, a user would choose the amount of CLOAG they would like to send CLOAGed to an address. CLOAGCoin would then work in the background to create a number of smaller Enigma transactions of an even amount, which can be CLOAGed and submitted to the CLOAG network over a set period of time. This batching process will be compatible with 'combined' Enigma transactions, providing further CLOAGing protection for transfers.

6. FAQ

Q. HOW DO CLOAGERS ASSIST AN ENIGMA TRANSACTION?

CLOAGers provide one or more inputs that are used to 'CLOAG' the input from the sender. CLOAGers also supply a series of return addresses which return their input and also reward the CLOAGer 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 Enigma transaction. The Enigma system will also check the target address so that 'CLOAGed' outputs mirror the true output as closely as possible.

Q. HOW LONG DO ENIGMA TRANSACTIONS TAKE TO COMPLETE?

Enigma transactions are currently allotted one minute to complete. CLOAGing nodes helping to 'CLOAG' an Enigma transaction will reserve the necessary funds until the Enigma transaction completes or the allotted time expires. In the case of an expired or aborted Enigma transaction, funds are unlocked locally for re-use.

Q. HOW DOES ENIGMA AFFECT STAKING?

Any coins used in a Enigma transaction (as a Sender or CLOAGer) will have their coin-age reset. It should be noted however, that participating in CLOAGing should provide a much higher return than staking. The CLOAG Team is working to revise the Enigma algorithm for the upcoming hard-fork release (Enigma 1.1). Please see Section 5 - 'The Future of Enigma – Further Development' for more details.

Q. DO I NEED A CERTAIN AMOUNT OF CLOAGS IN MY WALLET BALANCE TO BE A ENIGMA CLOAGER?

You can offer your services for CLOAGing regardless of the balance in your CLOAGCoin wallet. When Enigma CLOAGing is enabled, CLOAGCoin will reserve a portion of your balance for participating in Enigma CLOAGing, for which you will earn a CLOAGing 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 Enigma announcements by advertised CLOAGing balance. It should be noted that wallets with a higher balance have a higher chance of being chosen as a CLOAGer as they are more likely to have the required CLOAGing balance available for larger Enigma transactions.

Q. HOW DOES THIS PROTECT AGAINST A TIME BASED ATTACK WHERE SOMEONE LOOKS ON THE BLOCKCHAIN FOR IDENTICAL INPUTS AND OUTPUTS?

Enigma transactions group the outputs and are ensured to have multiple matching output amounts to ‘CLOAG’ the recipient’s output.

Q. CAN THE ORIGINATOR OF A ENIGMA TRANSACTION BE DETERMINED BY EXAMINING THE SCRIPT SIGNATURE TO DETERMINE SIGNING ORDER?

No. During the signing process, script signature order is randomized when combining the signatures. The sender and the participating CLOAGers do this.

Q. CAN AN EAVESDROPPER MONITOR THE NETWORK TO WATCH FOR OUTGOING ENIGMA TRANSACTIONS BEING SUBMITTED TO THE NETWORK TO DETERMINE THE TRUE SENDER?

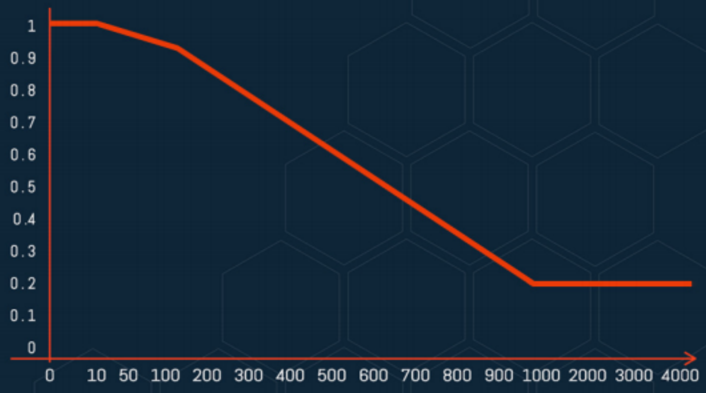
No. All parties in a random order submit an Enigma transaction to the network. This provides mitigation against such eavesdropping attacks.

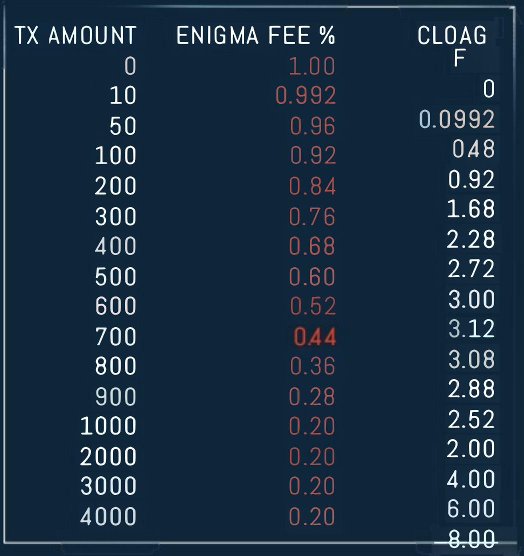
Q. WHAT IS THE FEE FOR AN ENIGMA TRANSACTION?

1% at 0 coins thru 2% at 1000 and higher coins. This is used to reward Enigma nodes that assist with CLOAGing an Enigma transaction. The fee is then mixed with the transaction and split between CLOAGers. It is not only a reward for participants but is used to help make determination of the transaction amount impossibly difficult. Each participant receives 80-120% of an equally split enigma transaction.

Q. HOW IS THE ENIGMA FEE DETERMINED?

The Enigma fee % is charged on a per transaction basis at these rates:





TX AMOUNT	ENIGMA FEE %	CLOAG F
0	1.00	0
10	0.992	0.0992
50	0.96	048
100	0.92	0.92
200	0.84	1.68
300	0.76	2.28
400	0.68	2.72
500	0.60	3.00
600	0.52	3.12
700	044	3.08
800	0.36	2.88
900	0.28	2.52
1000	0.20	2.00
2000	0.20	4.00
3000	0.20	6.00
4000	0.20	8.00

Q. DOES ENIGMA REQUIRE A HARD-FORK OF THE CLOAG NETWORK?

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

Q. WHAT IS THE MAXIMUM NUMBER OF CLOAGERS THAT CAN ASSIST IN A ENIGMA TRANSACTION?

The maximum number of CLOAGers is fixed at 25. The Enigma system is flexible and this number can easily be extended.

Q. HOW DOES ENIGMA PROTECT AGAINST 'BAD ACTORS'?

The Enigma system features extensive DDoS protection to 'blacklist' nodes for the duration of a session. If a Enigma node repeatedly refuses to sign, they will be excluded from Enigma CLOAGing invitations for the remainder of the current session. We are currently researching additional methodologies

for further penalizing uncooperative Enigma nodes and will likely implement a system that requires CLOAGers to escrow a nominal, refundable fee that could be claimed as a penalty in instances where a node attempts to block a Enigma transaction by refusing to sign the finalized transaction. It should be noted that whilst malicious nodes may attempt to hamper a Enigma transaction, they are not able to steal or misappropriate any funds.

Q. HOW ARE STEALTH AND ENIGMA TRANSACTIONS DETECTED/RECEIVED?

All incoming transactions are scanned. Stealth transactions are scanned for first (using the default ephemeral pubkey contained in a random OP_RETURN TX output). After this, Enigma transactions are then scanned for. Enigma transactions also use the standard ephemeral pubkey, but payments use an additional step involving a further derived key. Enigma outputs are generated using a hash of the ephemeral pubkey, a private stealth address hash and the output index.

When scanning for Enigma transactions, the zero-index payment addresses are generated for each owned stealth address [HASH(ephemeral_pubkey, hash_stealth_secret, 0)]. If a match is found for the zero-index of a stealth address, additional addresses are generated for the remaining indexes [num_tx_outputs] and these are scanned against to detect payments. See FindEnigmaTransactions in wallet.cpp for more info.

A similar scanning method is employed by CLOAGers prior to signing an Enigma TX to ensure they are getting reimbursed correctly. See GetEnigmaOutputsAmounts in wallet.cpp for more info.

7. 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] https://bitcointalk.org/index.php?topic=279249.0
(CoinJoin: Bitcoin Privacy for the Real World)

[06] https://bitcointalk.org/index.php?topic=27787.0
(Proof of Stake Instead of Proof of Work)

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

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

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

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