Muon: A Secure, Scalable, Private, Decentralized Digital Currency

muoncrypto@gmail.com

1. Introduction

When dealing with physical currency, the uniqueness of a note of fiat currency is established by a serial number, as well as a variety of anti-counterfeit measures such as watermarks, raised printing, and security fibers [1]. When dealing with electronic currency, however, serial numbers are easily copied, and one does not have the luxury of relying on physical anti-counterfeit measures. Furthermore, with a decentralized currency there is no central establishment to ensure that coins of a given serial number will not be copied.

The challenge of decentralized currencies is to somehow establish consensus that only one unique copy of any given coin exists, without relying on a trusted establishment. One way to do this is to have all nodes in the network keep a public ledger of ownership of all coins in circulation, which is maintained individually by each node in the network. Then, when a node wishes to make a transaction, this transaction is broadcast to the network, and assuming the node is spending money that it is authorized to spend, ideally every node in the network then updates their ledger to establish a new consensus of ownership. However, one must keep in mind that not all nodes in the network are necessarily trustworthy and some may intentionally or unintentionally sabotage the consensus process. Thus, the network must be able to establish consensus even when some of its nodes are dishonest. In mathematics this is often referred to as the Byzantine Generals' Problem [2], in which a group of generals communicating by messengers must decide unanimously whether to attack or retreat, even while some generals of the group may be traitors attempting to sabotage the consensus process.

To solve the Byzantine Generals' Problem and establish consensus, proof-of-work digital currencies such as Bitcoin discard serial numbers altogether and instead define each coin by a chain of digital signatures, in which ownership is transferred from one party to another [3]. To prevent dishonest nodes from inventing alternative ownership chains for coins, each transaction is validated in a block along with other transactions by the entire network solving a mathematical problem that is difficult to solve but easy to verify. The longest chain of ownership transaction blocks, i.e., the chain with the most proof-of-work, is then agreed to be the valid *blockchain*. For a dishonest set of nodes to propose an alternative chain of ownership, they would have to create a longer alternative blockchain, effectively overpowering the computing capacity of all of the honest nodes in the network. This attack is referred to as a 51% attack [4].

While proof-of-work currencies are now well-established, they also have many shortcomings that must be addressed in order for decentralized digital currencies to achieve the goal of worldwide adoption. These shortcomings include: Excessive power consumption, slow and unpredictable transaction processing times, unpredictable fees, limited throughput, and consistent movement toward centralization due to a variety of issues such as concentration of mining power, memory storage requirements, and concentration of influence by core software developers [4, 5, 6].

While there are a variety of proposed and implemented solutions to all of the previously mentioned issues, each of these so far comes with tradeoffs that this author views as unacceptable. Solutions to address scalability such as the Lightning Network [7] for Bitcoin come at the cost of centralization; alternative coins such as Ripple are explicitly centralized [8]; proof-of-stake currencies concentrate power into members with the largest currency holdings and concentrate security vulnerabilities [9]. Meanwhile, only a handful of digital currencies address the issue of privacy at all [4].

The four pillars of a decentralized digital currency, in this author's opinion, are:

1) **Security**: The network must be secure against double-spend attacks, and its users must be confident that when they receive funds, these funds will never disappear from their account.
2) **Scalability**: The network should securely process any individual transaction within a few seconds. The network should be capable of processing transactions asynchronously and without participation of the entire network. In a Muon network of N nodes, for example, only *O(log(N))* nodes participate in the validation of any single transaction request. Additionally, nodes should not have to maintain a history of all transactions.
3) **Privacy**: Knowledge of the sender, receiver, and transaction amount of a given transaction should be restricted only to the parties involved in the transaction. This protects the users of the network and guarantees the fungibility of its currency.
4) **Decentralization**: Each node in the network should have modest and equal memory, network, and CPU requirements for sending and receiving currency, and participating in the validation of transactions. Additionally, no node, user, or organization should have any more power or influence over the network than any other, regardless of their “stake” in the network.

The creation of Muon aims to resolve the shortcomings of other cryptocurrencies and provide satisfactory solutions to each of the four pillars, using a novel means of validating transactions.

2. Transactions

In Muon, individual transactions are validated with a chain of digital signatures, similar to the way proof-of-work currencies define a chain of ownership. However, proof-of-work is not required in Muon to validate individual transactions. Instead, for each transaction a set of nodes is chosen randomly from the available nodes in the network to be *auditors* for the node originating the transaction, which we will call the *originator*. To prevent the possibility of corruption, these auditors are not known in advance to the originator. Instead, the auditors are revealed once the originator broadcasts the transaction.

The originator includes a nonce n_i in its broadcast that is incremented with each transaction so that $n_{i+1}=n_i+1$ for consecutive transactions. The nonce is used to seed the election of auditors, and ensures that the network always agrees on the order of transactions, since nodes will simply ignore transaction broadcasts with nonces that are above or below their next expected value. Auditors validate a transaction by simply signing off on the transaction (using a hash of the nonce *and* the transaction in question) in an agreed-upon ordering that is established during the auditor election process, thus creating a signature chain. A transaction is only validated when all auditors agree that the transaction is valid. Transaction fees are small, constant percentages that are disbursed equally to all nodes participating in the network, and are used to encourage honest participation.

Finally, an originator may only spend up to half of its account per transaction (minus the transaction fee), meaning that each transaction must send at least half of its input back to the originator. This is to ensure that half of the originator's transaction is held as collateral in the event that its auditors detect a double spend attempt. If auditors detect a double spend attempt, they will broadcast a special punitive transaction in which the originator's collateral is taken as a transaction fee.

3. Double-Spend Prevention

A transaction is said to be *valid* if the originator is spending Muons that it is authorized to spend (i.e., Muons it has received but not yet spent), and is said to be *validated* if the transaction is signed by an auditor chain.

Let us suppose that a dishonest node now wishes to attempt to double-spend. If the dishonest node broadcasts two valid but conflicting transactions using the same nonce, auditors will be elected according to the nonce value, and the first transaction to reach the first auditor will be the only transaction accepted by the network, since the first auditor will sign the first valid transaction that it receives, pass it on to the next auditor, and remembering its previous transaction, will not sign any more transactions that contain the same nonce.

If on the other hand, a dishonest node broadcasts valid but conflicting transactions using different nonces, the network will ignore the out-of-order nonce transaction until the first transaction is processed. Then, assuming the out-of-order nonce is one greater than the nonce of the previous transaction, new auditors will be elected and will determine whether the new transaction is valid. Since at that point the network has already reached consensus on the first valid transaction, auditors will reject the conflicting transaction. Thus, regardless of the nonce values chosen, dishonest nodes will not be able to double-spend.

Now, suppose a dishonest node wishes to double-spend by creating a forgery of its last audited transaction, only with the transaction sending money to different recipients. Because the auditors' signatures include a hash of both the nonce and the transaction, their signatures cannot be forged. Another method a dishonest node might try is to forge an audited double-spend transaction by creating a set of dishonest auditors that will be chosen by a given nonce. However, the method used by Muon for electing auditors precludes the possibility of an originator planting its auditors.

Finally, suppose an attacker has corrupted proportion *p* of the network, so that each auditor has a probability *p* of allowing the attacker to double-spend. In order for the attacker to successfully double-spend, all C auditors chosen must be corrupt, where C is the auditor chain length. Thus, in order for the attacker to have a positive expectation of profit, the proportion of the network corrupted must be $p > \left(\frac{1}{2}\right)^{\frac{1}{C}}$. With five auditors, for example, an attacker must control 87% percent of the network. To prevent a *Sybil attack*, in which an attacker creates many nodes to increase the proportion of the network it controls, nodes are required to maintain a *minimum account balance* to be eligible to audit transactions and receive disbursements.

While Muon is designed to disallow double-spending, it is also predicated on a principle of zero operational risk. Thus, there is no mechanism for altering or rolling back transaction histories. In the event of a successful double-spending attack, new money will be created and all Muon holders will absorb some depreciation of their assets in the form of inflation. This is preferable to each individual currency holder risking a loss of funds after receiving Muons, since such risk undermines confidence in the legitimacy of all transactions. When transactions are validated, recipients can be assured that the Muons they received are theirs to spend.

4. Auditor Election Process

Ideally, most of the work for electing auditors is done before the originator broadcasts a transaction request. To do this, for each node we divide the network with N nodes into roughly N/log(N) *endorsement groups*, with each endorsement group containing a *potential auditor*. If a potential auditor meets certain criteria, it is qualified to present itself as an *endorsed auditor*, which it proves by collecting endorsement signatures from its group. When an originator broadcasts a transaction, the endorsed auditors reveal themselves and choose C auditors among themselves to be the auditors for the transaction, where C is the auditor chain length. Each step of the process is deterministic, so that the same auditors will be chosen if an originator broadcasts two conflicting transactions with the same nonce.

Communication within endorsement groups is conducted via *selective broadcasts*, which are broadcasts that are encrypted using the public keys of all members of the group, so that nodes outside the group cannot eavesdrop on the auditor selection process. While there is the possibility that some members of the group will be spies for a dishonest node, the network can tolerate a certain amount of spying because it has many endorsement groups and relatively few of them will contain endorsed auditors.

Suppose we have a network of N nodes that we wish to divide into N/log(N) endorsement groups. List all nodes in numerical order by their public keys, and then shuffle the list using $H(n_{tx} \| K_o)$ as a seed, where n$_{tx}$ is the transaction's nonce, and K$_o$ is the originator's public key address. To the enumeration of the shuffled list apply $mod\left(\left\lfloor \frac{N}{\log(N)} \right\rfloor\right)$ to create approximately N/log(N) endorsement groups of size log(N).

For a transaction with nonce n_{tx}, each node in each endorsement group selectively broadcasts a signature $s_i = E(H(n_{tx} \| K_o), k_i)$, where

E =: a public key encryption method,
H =: a hash function,
k =: a private key, and
K$_o$ =: the public key of the originator.

Such signatures produce pseudorandom numbers that are impossible for an originator to know in advance, but easy to verify. Within a group of size G, nodes compare their signatures to establish an ordering $\langle s_i \rangle$ of signatures, then elect the node that satisfies $\min_{i \in G} |s_i - H(\langle s_i \rangle)|$ as the potential auditor for the group.

A potential auditor is qualified to become an endorsed auditor if $(s_i + H(\langle s_i \rangle))\, mod\left(\left\lfloor \frac{N}{\log(N)} \right\rfloor\right) \leqslant \log(N)$. The purpose of this threshold criteria is to reduce the number of endorsed auditors in order to speed up the transaction auditing process, but it is not strictly necessary for Muon to function. If a potential auditor is qualified to become an endorsed auditor, the potential auditor keeps its group's signatures as proof of endorsement, to be provided in the event of a transaction request.

Using the above method, all nodes within an endorsement group should agree on which node is the group's auditor, as long as they agree on all members of the group. Disagreements in membership are resolved by the liveness protocols, which are described in section 6. In the very unlikely event that not enough potential auditors meet the threshold to be promoted to endorsed auditors, the network can temporarily raise the threshold enough to promote C potential auditors.

When a transaction is requested, all endorsed auditors broadcast their signatures along with their proof of endorsement, and choose the C lowest ranked auditors among themselves to be auditors for the transaction, using each auditor's $s_i + H(\langle s_i \rangle)$ as the ranking metric.

5. Privacy

In theory, Muon could support a simple protocol for tracking balances by simply having each node maintain a list of the balances of all accounts, and update relevant balances after each transaction. This method has the advantage of not requiring memory storage for transaction history; however, it is incompatible with maintaining the privacy of Muon currency holders.

Thus, to establish privacy, Muon balances are maintained indirectly via transactions. A validated Muon transaction consists of a nonce, a set of *outputs* consisting of recipient addresses with corresponding amounts to send, and a set of *inputs* consisting of one or more validated transactions whose outputs contain the originating node's address.

All nodes in the network maintain a set of *unspent transaction outputs* (UTXO), which is used to determine whether an originator is authorized for the Muons it is attempting to spend in any transaction. Although there is nothing to prevent nodes from maintaining transaction histories, only the UTXO is required. After each transaction, each node updates its UTXO to remove spent outputs from the originator and add unspent outputs to its recipients. Precautions against potential uncontrolled growth of the UTXO are discussed in section 7.

Maintaining balances via transactions allows the Muon network to apply the same techniques as cryptocurrencies such as Monero [4] to ensure the privacy of its users and the fungibility of Muons. To maintain the privacy of senders, the identity of the originator is grouped with other potential addresses via ring signatures. To maintain the privacy of recipients, transaction outputs are listed via stealth addresses. Transaction amounts themselves are obscured via techniques such as Pedersen commitments [10] and range proofs [11], so that nodes other than those involved in the transaction are only aware that authorized funds are being spent, and that the transaction inputs and outputs are balanced.

6. Liveness

Nodes must be continuously live for a minimum liveness time threshold before they qualify to receive transaction fee disbursements. This incentivizes nodes to remain live and support the Muon network. Nodes joining the network broadcast a *join request*. The join request is then audited as a special transaction, and the joining node is assigned a list of neighbors. A node's network join date is based on the first timestamp of liveness on record by the first auditor of the join request.

To guarantee the performance of network transactions, nodes are required to remain continuously live, and meet certain minimum network responsiveness standards as measured by communication with their neighbors. Each node is required to have a certain minimum number of neighbors to guarantee the connectedness of the network.

Nodes communicate liveness to their neighbors through *heartbeats*, which are short signed messages with a timestamp. Nodes measure the responsiveness of their neighbors by calculating the time elapsed when receiving heartbeat timestamps. If a node determines that one of its neighbors is lagging in performance (which we refer to as a *questionable node*), it may sign a petition that is then circulated among the questionable node's neighbors. If the petition is signed by a supermajority of the questionable node's neighbors, these nodes can present the petition as a node *departure request* to remove the questionable node from the network. After the departure request is audited, the questionable node has a set period of time to contest the request by broadcasting a *remain request*.

Remain requests are automatically granted; however, only a certain number of remain requests can be audited for a questionable node within a certain time period. If the number of remain requests exceeds a certain threshold in a given time period, the remain request is not granted and the questionable node is removed from the network membership list. If a departure request is not contested, then the questionable node is removed from the membership list.

7. Disbursements

All nodes supporting the network begin to receive disbursements from transactions fees after they have been continuously live for the duration of the minimum liveness time threshold, and meet the minimum account balance requirement. Transaction fees are fixed at a low rate and are disbursed from the nodes originating the transaction. Since Muon does not require miners, there is no need to have a schedule for the creation of new Muons. The total amount of Muons in circulation is fixed at genesis.

Transaction fees incentivize nodes to support the network by auditing transactions, and help manage the size of the UTXO. Fees apply to each output of a transaction, since these increase the size of the UTXO, and are subtracted for each input as long as the net fee is nonnegative. For example, the sender of a transaction with one input and two outputs (one to the recipient, one as change back to the sender) pays two transaction fees and is paid one transaction fee, so overall the sender pays one net transaction fee. Therefore, transactions that decrease the size of the UTXO are free, while the cost of transactions that increase the UTXO is proportional to the number of additional outputs.

To further manage the size of the UTXO and prevent the accumulation of transaction “dust”, i.e., very small unspent outputs, we introduce a minimum transaction fee f_{min}. For a regular transaction from an input amount I with a fee f where $f<\frac{I}{2}$, a maximum of $\frac{I}{2}-f$ can be sent to any recipient, while f is disbursed and $\frac{I}{2}$ is sent back as change to the sender. If $I<2f_{min}$, then the entire input is disbursed as a transaction fee, thus there is no incentive to spend the input. These inputs are therefore automatically removed from the UTXO and disbursed.

Now, suppose we wish to transfer the entirety of a Muon account, paying the minimum amount of fees. With a minimum transaction fee in place, the minimum number of transactions n necessary to empty an account occurs when $\frac{I}{2^n}>2f_{min}$, so $n=lg\left(\frac{I}{f_{min}}\right)-1$. The transaction fee for an input I is given by $f(I)=\frac{kI}{2}$, where k is a constant fraction of the input, e.g., 0.001. The total amount of transaction fees paid to empty an account is therefore given by

$F(I)=kI\left(\frac{1}{2}+\left(\frac{1}{2}\right)^2+...+\left(\frac{1}{2}\right)^n\right)+2f_{min}=kI\left(1-\left(\frac{1}{2}\right)^n\right)+2f_{min}$. Substituting $n=lg\left(\frac{I}{f_{min}}\right)-1$, we have $F(I)=kI+2f_{min}(1-k)$. Thus, we are assured that the total fee paid is a linear function of the input to transfer, and will be a reasonable amount even for large inputs.

8. Network Partitions

Because all nodes in the network are aware of the number of nodes online, nodes can prevent an attacker from partitioning a small part of the network to carry out double spends by refusing to accept transactions validated by a small subset of the network's membership at a given time.

In the event that the network is severed into large pieces by a catastrophic split, e.g., U.S. nodes being temporarily split from Chinese nodes, transactions processed in each separate network will be valid, and the networks will be able to combine their UTXO's when the network merges again. Because there is no mechanism for rolling back transactions, there is a possibility for double-spending to occur during a catastrophic split, although this would require an attacker to have simultaneous access to both partitions of the network while no nodes in the network itself have such access.

9. Software Updates

For the long-term sustainability of the network, it is inevitable that changes such as re-adjustments of hardcoded constants or security patches may need to be implemented. To prevent a small group of software developers from monopolizing control of the network, software updates may be proposed by any node at any time; all software updates must be ratified by a supermajority of the nodes online during three separate votes, each no less than 24 hours apart, subject to the same security precautions used against network partition attacks.

10. Conclusion

A new model for cryptocurrency is presented, aiming to satisfy the four most important pillars of digital currency: Security, scalability, privacy, and decentralization. A novel means of validating transactions is presented, using signature chains of nodes serving as auditors chosen prior to each transaction to prevent double-spending attacks. The model is compatible with maintaining privacy of transactions. To ensure scalability, the model does not require any node in the network to maintain a full history of transactions; instead, only storage of unspent transaction outputs is necessary. Disbursements through transaction fees are used to manage the size of the unspent transaction output set as well as provide incentive for nodes to support the network.

11. References

1. https://d39pc38av48c2g.cloudfront.net/sites/default/files/security/pdf/100_2013_Features.pdf
2. L. Lamport, R. Shostak, M. Pease, The Byzantine Generals Problem https://people.eecs.berkeley.edu/~luca/cs174/byzantine.pdf
3. S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System https://bitcoin.org/bitcoin.pdf
4. Nicolas van Saberhagen, CryptoNote v 2.0 https://cryptonote.org/whitepaper.pdf
5. https://kb.myetherwallet.com/gas/what-is-gas-ethereum.html
6. https://bitcoin.org/en/development
7. https://lightning.network/
8. Ripple Explained with David Schwartz, Chief Cryptographer of Ripple Labs https://www.youtube.com/watch?v=GyNXedeCyNg
9. Proof of Stake FAQs https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs
10. T. Pedersen, Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing https://link.springer.com/content/pdf/10.1007%2F3-540-46766-1_9.pdf
11. A. Gibson, An investigation into Confidential Transactions https://github.com/AdamISZ/ConfidentialTransactionsDoc/blob/master/essayonCT.pdf

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBFuviVEBCADAkn0fWosRwbrk8gkux2BruANCf2F9fgX8X/BoU4R105l/93Ry
hxhpu7uu+PsXjMk56DoOpdnw5DUBjBtC2l3nu0qpyt4kHmJgTRwAt8tcgklF3B+O
V9wUiucQUjMDbINXyNBC000IpC+KyhFRzYukIPJg1OAdHr6W06PjfNtK072JxLjt
gI8kfP0RGDFiW9JEOJoC76GLbaCrKZo6s0V9bvlgq4ktHdIXDwmifIHQ5k8yneIo
G2C2eTqzR4VDrg3O7+UCJ1F3JiEnmh7SwUaT2XbpkDO2OhbDAvDvpZpo3xvKuXq/
8r/bWiJhCxyl+lKbchYO50XpW3nrj3wgvPnXABEBAAG0Kk11b24gQ3J5cHRvY3Vy
cmVuY3kgPG11b25jcnlwdG9AZ21haWwuY29tPokBVAQTAQgAPhYhBELdgN0ubL4M
AWw6L9mQ5r2vMPuWBQJbr4lRAhsDBQkDwmcABQsJCAcCBhUICQoLAgQWAgMBAh4B
AheAAAoJENmQ5r2vMPuWw3IIAKBsv059q0tSISSOo8Gb+Fz+OrvooqyqiKF+Y/zx
OFaYVUxG4Co/2MxX0W6QGJBBA2D+I7CRkv2Rmfqp5lmAe+d/rNhuV1rlIQr1CqU8
Lz+PZHgnizK0NVH/K9yAMeb1xYulX2r33Ogu3ln6FkhqQI4Y3Xy89UQP4UuGiR9s
Jn9UrfntU16F/4pPkLBjM2gOXarlKUQUI83fQ8vZ0rc33UX/2fVy6GBxyFJEkzeN
h/8R0d3HCGRxW2rkgnC2G3UjI2KO2ss3SLKxt12/a5QTBmQ7AwJseMWip7cqRWDj
dckIUOCM4G8ovs7Ve3AHrQGJMiHrPBX8dnE2R5oz3sI2RGa5AQ0EW6+JUQEIAKaa
5gWoZHpW8alxWXMpR4C4g0OpldrUvzCnytSr46ZXEg7OXkckujwyd/FdCKV9OHQa
H+2+ke0XB3LBFXB2IeDRbD7kAO+SGbVgQ/AFYuXn7C6HNx/go01WK15+USqm3Ssg
/3A+l9BU0TX6RGqbQtTP0bJoVbW5AN6e6IAkhiXml8umPGpyVWvMOtRu4T3IaB6N
XhIGP6tfGSNcUSP1N+KMQHf/QUwBoNTkQZenICvJ6dq9eFmd2PXg/YjH4GVbvV6K
veqseJNpv6oksBu2tk2DqOEoEqzkpBRX1bZjfKW+QDKw3nf4CJXkN7++8nI1AUcL
K4FYhwagI9YMZd06Q1sAEQEAAYkBPAQYAQgAJhYhBELdgN0ubL4MAWw6L9mQ5r2v
MPuWBQJbr4lRAhsMBQkDwmcAAAoJENmQ5r2vMPuW6nAH/2QQEsJVolJjWV5rT3aM
8b1alvOi6zfF3e9pDjj3ZfP2e5RNW+6LlVgoE4kPyaTFqemAFkiQ+pJ1ITqbZA1p
4Lxgy+SR8xrQfS2S5vE6ruCdW1eVHYzwYTi0Us7r/nvNYkIKTOEQByhmqdCRL1zz
hUp/UW15wDpZ8xAGcXmP86A6BX+O0ZWmpPrLWB3/8rDRH+0yjRuJlz1wBe374/SZ
sNzpj4PJ9plJZGCHHI+c1pYT1KTciWbMZ3LMuJ2CsiYCvmNVScoxcu/5BjOdsi9i
O4Gh5NdaJu1aBeOxX0OIbQFlNwoe6l/AKYBjkudJ3eI7pOxaNrG6RpzskQyiE7Eh
F7w=
=nfFh
-----END PGP PUBLIC KEY BLOCK-----
```

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Muon: A Secure, Scalable, Private, Decentralized Digital Currency
-----BEGIN PGP SIGNATURE-----

iQFJBAEBCAAzFiEEQt2A3S5svgwBbDov2ZDmva8w+5YFAluyUUIVHG11b25jcnlw
dG9AZ21haWwuY29tAAoJENmQ5r2vMPuWEUEIAJsKqnCb5GxssMl5WfR9xjbsMrr4
OEJBzSiwBVbtZuJu4LQs0BlFBOAgfbMiOVmeg/Mzq8639ulz70J0xLgiHf9cqCqW
/KSs6NT4Cnm1dkJSNLEEjfOOSAGQubAuUBQxRP+lsbiLI31O1hjX3pIfeBTIlRrJ
2x6cLq6oVnd3BCtZ4khsc01Q7lHemyNM29Ck8TDXflwaomfJUwQ/CZLcYg6xgiYe
z9iNU49gGsJuZ9RqEGHMljeq4z8oT05ba8061TEq/piZzzjXfs6k/oo77JGisljd
kwxqsBaeF6/KNgLBhAVqnbRvezGnQeYDwAE+ReC1nBLoZl4rU1wC5YZh8S4=
=mS02
-----END PGP SIGNATURE-----