Stellite: A modern, secure and decentralized cryptocurrency.

Jan 2018

EXECUTIVE SUMMARY

In a few paragraphs, we are explaining the implementation methodologies and the technologies used for making Stellite.

Node list distribution

Most cryptocurrency including bitcoin, monero etc, hardcode trusted DNS nodes and also trusted peers for helping users set up their own nodes, this has been proven to work but it is calls for some security risk. What would happen if a hardcoded node or DNS is hacked? It'll cause all the new nodes from that point on to follow a different Blockchain ledger. We propose a solution to this. We publish a node list, on a decentralized network called IPFS from where the new node downloads the ledger. This node list can be voted on by users by hosting it. The one which is hosted the most can be used as the default one in the node's core. This leads to the cryptocurrency being truly decentralized.

Complete node anonymity

Nodes are not private, it runs on a specific port and using programs like nmap or masscan an attacker can easily scan for nodes and take them down using vague requests. We will provide a running tor instance with which a user of a node can configure a node in such a way that people can't easily identify it being a node without compromising the tor network first.

True mining for all devices

Devices such as smartphones and telephones have been talked about intensively but has not had a proper implementation yet(Currencies like electroneum has made mobile mining experience and not a real mobile miner). We are proposing an idea where even the smallest of devices can provide hashes to make the network stronger than ever. We do this by providing the user with a mobile, web browser miner so that he/she can mine with N number of devices to maximize their productivity.

The math for this is simple, a mid tier smartphone provides around 23H/s at full throttle, when at 25% usage provides around 6H/s. A user can use her specific wallet address 'X' to mine on as many devices as possible. So if Alice has 200 devices she has 6 * 200 = 1200 Hashes or 1.2KH/s. Which is equivalent to 2 GTX 1080 graphics card running at full potential.

This is possible due to a variable difficulty algorithm we have worked on, if a device such as a smartphone can only provide 10H/s the improved pool server immediately gives out block templates exactly corresponding to that hashrate improving not just speed but efficiency too. The new pool works on an event loop and also has adequate proxy servers for running millions of workers. This will not just strengthen the network but also redistributes wealth globally.

CryptoNote Algorithm

The CryptoNote algorithm is released under an open source license and has been adopted and incorporated into Stellite as it forms the basis for a solid, well tested cryptocurrency application. It is the same technology used by some of the best currencies out there like monero and bytecoin. Now we can discuss some of the merits of the currency as well as cryptonote.

Untraceable payments

The ordinary digital signature (e.g. (EC)DSA, Schnorr, etc...) verification process involves the public key of the signer. It is a necessary condition, because the signature actually proves that the author possesses the corresponding secret key. But it is not always a sufficient condition.

Ring signature is a more sophisticated scheme, which in fact may demand several different public keys for verification. In the case of ring signature, we have a group of individuals, each with their own secret and public key. The statement proved by ring signatures is that the signer of a given message is a member of the group. The main distinction with the ordinary digital signature schemes is that the signer needs a single secret key, but a verifier cannot establish the exact identity of the signer. Therefore, if you encounter a ring signature with the public keys of Alice, Bob and Carol, you can only claim that one of these individuals was the signer but you will not be able to pinpoint him or her.

This concept can be used to make digital transactions sent to the network untraceable by using the public keys of other members in the ring signature one will apply to the transaction. This approach proves that the creator of the transaction is eligible to spend the amount specified in the transaction but his identity will be indistinguishable from the users whose public keys he used in his ring signatures.

It should be noted that foreign transactions do not restrict you from spending your own money. Your public key may appear in dozens of others' ring signatures but only as a muddling factor (even if you already used the corresponding secret key for signing your own transaction). Moreover, if two users create ring signatures with the same set of public keys, the signatures will be different (unless they use the same private key).

Unlinkable Transactions

Normally, when you post your public address, anyone can check all your incoming transactions even if they are hidden behind a ring signature. To avoid linking you can create hundreds of

keys and send them to your payers privately, but that deprives you of the convenience of having a single public address.

Stellite's CryptoNote solves this dilemma by an automatic creation of multiple unique one-time keys, derived from the single public key, for each p2p payment. The solution lies in a clever modification of the Diffie-Hellman exchange protocol. Originally it allows two parties to produce a common secret key derived from their public keys. In our version the sender uses the receiver's public address and his own random data to compute a one-time key for the payment.

The sender can produce only the public part of the key, whereas only the receiver can compute the private part; hence the receiver is the only one who can release the funds after the transaction is committed. He only needs to perform a single-formula check on each transactions to establish if it belongs to him. This process involves his private key, therefore no third party can perform this check and discover the link between the one-time key generated by the sender and the receiver's unique public address.

An important part of our protocol is usage of random data by the sender. It always results in a different one-time key even if the sender and the receiver both remain the same for all transactions (that is why the key is called "onetime"). Moreover, even if they are both the same person, all the one-time keys will also be absolutely unique.

Double-spending proof

Fully anonymous signatures would allow spending the same funds many times which, of course, is incompatible with any payment system's principles. The problem can be fixed as follows.

A ring signature is actually a class of crypto-algorithms with different features. The one Stellites's CryptoNote uses is the modified version of the "Traceable ring signature". In fact we transformed traceability into linkability. This property restricts a signer's anonymity as follows: if he creates more than one ring signature using the same
private key (the set of foreign public keys is irrelevant), these signatures will be linked together which indicates a double-spending attempt.

To support linkability, stellite's CryptoNote introduced a special marker being created by a user while signing, which we called a key image. It is the value of a cryptographic one-way function of the secret key, so in math terms it is actually an image of this key. One-wayness means that given only the key image it is impossible to recover the private key. On the other hand, it is computationally impossible to find a collision (two different private keys, which
have the same image). Using any formula, except for the specified one, will result in an unverifiable signature. All things considered, the key image is unavoidable, unambiguous and yet an anonymous marker of the private key.

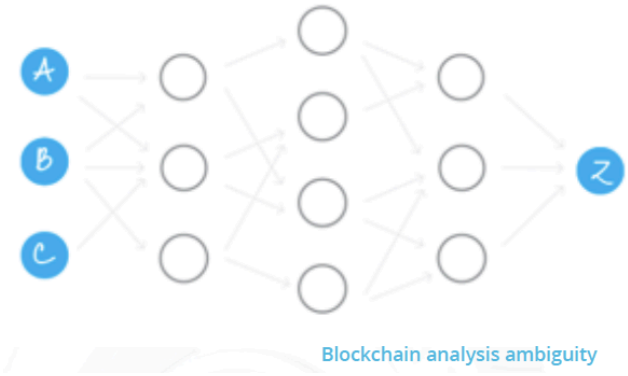
All users keep the list of the used key images (compared with the history of all valid transactions it requires an insignificant amount of storage) and immediately reject any new ring signature with a duplicate key image. It will not identify the misbehaving user, but it does prevent any double-spending attempts, caused by malicious intentions or software errors.

Blockchain analysis resistance

There are many academic papers dedicated to the analysis of the Bitcoin's blockchain. Their authors trace the money flow, identify the owners of coins, determine wallet balances and so on. The ability to make such analysis is due to the fact that all the transfers between addresses are transparent: every input in a transaction refers to a unique output. Moreover, users often re-use their old addresses, receiving and sending coins from them many times, which simplifies the analyst's work. It happens unintentionally: if you have a public address (for example, for donations), you are sure to use this address in many inputs and transactions.

Stellite's CryptoNote is designed to mitigate the risks associated with key re-usage and one-input-to-one-output tracing. Every address for a payment is a unique one-time key, derived from both the sender's and the recipient's data. It can appear twice with a probability of a 256-bit hash collision. As soon as you use a ring signature in your input, it entails the uncertainty: which output has just been spent?

Trying to draw a graph with addresses in the vertices and transactions on the edges, one will get a tree: a graph without any cycles (because no key/address was used twice). Moreover, there are billions of possible graphs, since every ring signature produces ambiguity. Thus, you can't be certain from which possible sender the transaction edge comes to the address-vertice. Depending on the size of the ring you will guess from "one out of two" to "one out of a thousand". Every next transaction increases the entropy and creates additional obstacles for an analyst.



Blockchain analysis ambiguity

Standard CryptoNote transactions

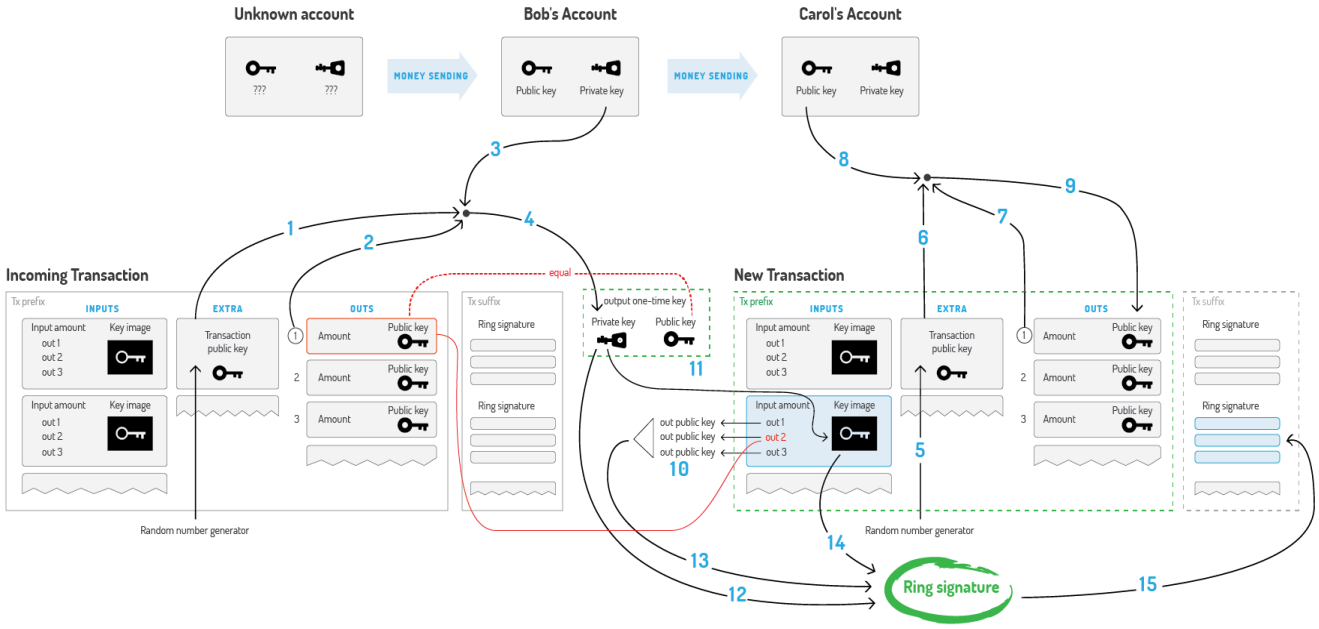
A standard stellite CryptoNote transaction is generated by the following sequence covered in the white paper.

Bob decides to spend an output, which was sent to the one-time public key. He needs Extra (1), TxOutNumber (2), and his Account private key (3) to recover his one-time private key (4).

When sending a transaction to Carol, Bob generates its Extra value by random (5). He uses Extra (6), TxOutNumber (7) and Carol's Account public key (8) to get her Output public key (9).

In the input Bob hides the link to his output among the foreign keys (10). To prevent double-spending he also packs the Key image, derived from his One-time private key (11).

Finally, Bob signs the transaction, using his One-time private key (12), all the public keys (13) and Key Image (14). He appends the resulting Ring Signature to the end of the transaction (15).



A Standard stellite cryptonote transaction.

Adaptive Limits

A decentralized payment system must not depend on a single person's decisions, even if this person is a core developer. Hard constants and magic numbers in the code deter the system's evolution and therefore should be eliminated (or at least be cut down to the minimum). Every crucial limit (like max block size or min fee amount)
should be re-calculated based on the system's previous state. Therefore, it always changes adaptively and independently, allowing the network to develop on it's own.
Stellite's CryptoNote has the following parameters which adjust automatically for each new block :-

1. Difficulty. The general idea of our algorithm is to sum all the work that nodes have performed during the last 720 blocks and divide it by the time they have spent to accomplish it. The measure of the work is the corresponding difficulty value for each of the blocks. The time is calculated as follows: sort all the 720 timestamps and cut-off 20% of the outliers. The range of the rest 600 values is the time which was spent for 80% of the corresponding blocks.
2. Max block size. Let MN be the median value of the last N blocks sizes. Then the "hard-limit" for the size of accepting blocks is 2*MN. It averts blockchain bloating but still allows the limit to slowly grow with the time if necessary. Transaction size does not need to be limited explicitly. It is bounded by the size of the block.

Smooth Emission

The upper bound for the overall amount of all digital coins is also digital :

MSupply = 264 − 1 atomic units
This is a natural restriction based only on the implementation limits, not on intuition like "N coins ought to be enough for everybody". To make the emission process smoother stellite's CryptoNote uses the following formula for block rewards :

BaseReward = (MSupply − A) >> 18
Where A is amount of previously generated coins. It gives a predictable growth of the money supply without any breakpoints.

Egalitarian proof of work

The proof of work mechanism is actually a voting system. Users vote for the right order of the transactions, for enabling new features in the protocol and for the honest money supply distribution. Therefore, it is important that during the voting process all participant have equal voting rights. stellite's CryptoNote brings the equality with an egalitarian proof-of-work pricing function, which is perfectly suitable for ordinary PCs. It utilizes built-in
CPU instructions, which are very hard and too expensive to implement in special purpose devices or fast memory on-chip devices with low latency.

We propose a new memory-bound algorithm for the proof-of-work pricing function. It relies on random access to a slow memory and emphasizes latency dependence. As opposed to scrypt, every new block (64 bytes in length) depends on all the previous blocks. As a result a hypothetical "memory-saver" should increase his calculation speed exponentially.

Our algorithm requires about 2 Mb per instance for the following reasons :

1. It fits in the L3 cache (per core) of modern processors, which should become mainstream in a few years;
2. A megabyte of internal memory is an almost unacceptable size for a modern ASIC pipeline;
3. GPUs may run hundreds of concurrent instances, but they are limited in other ways: GDDR5 memory is slower than the CPU L3 cache and remarkable for its bandwidth, not random access speed.
4. Significant expansion of the scratchpad would require an increase in iterations, which in turn implies an overall time increase. “Heavy” calls in a trust-less p2p network may lead to serious vulnerabilities, because nodes are obliged to check every new block’s proof-of-work. If a node spends a considerable amount of time on each hash evaluation, it can be easily DDoSed by a flood of fake objects with arbitrary work data (nonce values).

One of the proof-of-work algorithms that is in line with our propositions is CryptoNight. It is designed to make CPU and GPU mining roughly equally efficient and restrict ASIC mining.