Stealthsend Whitepaper Brief

HONDO | ***stealthcoin***

September 09, 2014

Abstract

„Here, I describe the essential theory and properties of ***stealthsend***.
A more detailed description will accompany its launch."

– Hondo –

stealthsend

1 Introduction

CryptoNote provides exceptional anonymity for cyrpto-currencies. Its system of ring signatures removes certainty about the sending address, with the uncertainty being inversely proportional to the number of keys in the ring. In short, if there are N keys in a ring, then any key in the ring has a $1/N$ probability of ownership of the transaction, where ownership means the ownership of the private key authorized to sign the payment. For example, if 100 keys comprise a ring, then the probability that any key in the ring signed the transaction is 1%.

Despite the extraordinary technical achievement that CryptoNote represents, the ring signature system it utilizes poses serious limitations to its long term usefulness. Importantly, to validate a ring signature, the equivalent of one signature per key must be stored in the block chain so that the ring signature may be later validated. Additionally, the signing keys must be stored with the signature. For base64 encoded data, these requirements equate to 97 bytes per key. For example, if 100 keys are used in the ring, the storage burden for a transaction with a single input and single output is 9.5 kB (where 1 kB equals 1024 bytes).

2 Chandran Signatures

Undoubtedly, some refinements to the CryptoNote ring signature system could reduce this storage burden. To address the problem of block chain bloat, ***stealthsend*** will use a signature system described by Nishanth Chandran, $et\ al.$ [1], which is termed herein as „*Chandran signatures*". Chandran signatures reduce the storage burden from $\mathcal{O}(N)$ to $\mathcal{O}\left((N+12\sqrt{N})/3\right)$. For example, instead of requiring 9.5 kB, a 100 key Chandran signature (with no further optimizations) would would require about 7 kB.

Chandran signatures achieve this space savings by including non-interactive witness-indistinguishable (NIWI) proofs in the signature. These proofs commit to only a subset of the complete multisignature, requiring storage for just the subset. For each signature, the proofs require about 6ν storage, where $\nu = \sqrt{N}$.

3 Nonce Key Selection

Although Chandran signatures provide a significant savings in storage compared to CryptoNote ring signatures, it is possible to achieve further savings by selecting keys using a system of random nonces.

In the ***stealthsend*** system, two nonces are needed. The first nonce ϕ is used to establish the start of a range of public keys and the second, ψ, is used to select the desired number of keys from the range. Beyond these two nonces, requiring only eight bytes each, the size J (8 bytes) of the range needs to be specified, as well as N (8 bytes).

This system, described in more detail below, reduces the storage requirements for the signature keys from $\mathcal{O}(N/3)$ to essentially $\mathcal{O}(0)$. Thus, the overall storage requirements from CryptoNote ring signatures reduces from $\mathcal{O}(N)$ to $\mathcal{O}(4\sqrt{N})$. For example, a 100 key ***stealthsend*** signature requires only about 3.8 kB, compared to 9.5 kB for CryptoNote ring signatures. Thus, for a typical multisignature of 100 keys, ***stealthsend*** signatures require about 1/3 of the space of CryptoNote signatures.

To select the keys, a cryptographically random nonce, ϕ, is generated to seed a deterministic random number generator (PRNG), such as the Mersenne twister. The PRNG selects a number between 0 and the size of the entire keyspace (all the public keys found in the ***stealthsend*** blockchain). The range has boundries of this index I and $I+J$, where J is the size of the desired range. Keys are selected from the block chain by indices assigned according to their first appearance in the block chain. If the sender's public key, K_p is within the range, the nonce ϕ is kept along with J. If the K_p is not within this range, a new random nonce ϕ is generated.

Once an acceptable ϕ has been found, a second cryptographically random nonce ψ is generated and used to seed the PRNG. This PRNG is then used to select a set of N keys from the range specified by the nonce ϕ. If K_p is in this set, then the nonces, J, and N are stored with the signature created from them, otherwise a new nonce ψ is generated.

3 StealthSend Specifications

Although ***stealthcoin*** is a proof-of-stake (*PoS*) coin, ***stealthsend*** can not be *PoS* because *PoS* requires proof of ownership for the stake, which compromises anonymity unless memory expensive proofs are included with every staking transaction.

Although such proofs are theoretically possible, they are unnecessarily impractical for a crypto-currency. Therefore, ***stealthsend*** will be a proof-of-work coin, which will most likely use the low energy scrypt hashing algorithm.

Holders of ***stealthcoin*** will be able to convert to ***stealthsend*** at a ratio of 1:1 using a proof-of-burn conversion. This conversion will account for 85% of the StealthSend money supply. The other 15% of the ***stealthsend*** money supply will be emitted using a smooth emission algorithm, such that 1/2 of the remaining money supply is emitted every 2 years. The total money supply will depend on how much StealthCoin is converted to ***stealthsend***.

Finally, ***stealthsend*** will have a six minute block time.

[1] Chandran, N, $et\ al.$ Ring Signatures of Sub-linear Size without Random Oracles.