

# Trezarcoin

Sven van Gelder<sup>1</sup> and Oliver Ziegler<sup>2</sup>

<sup>1</sup>Sven.van.Gelder@web.de

<sup>2</sup>o.zieglerZ@web.de

## Abstract

Introducing Trezarcoin an extremely secure and energy efficient cryptocurrency, which combines the best features of both, Proof-of-Work and Proof-of-Stake mining. It's central innovation, 0% Proof-of-Stake is by far the most advanced implementation to date and is developed by Ghostlander. Trezarcoin supports multiple algorithms to achieve extra security. With this features Trezarcoin is targeted at users that value security highly.

## 1 NeoScript

NeoScript is a further development of Script and is aiming to increase security and improve performance on general purpose computer hardware while maintaining comperable costs and requirements. Although a very innovative design back in time, Script has developed certain vulnerabilities. The first announced differential cryptanalysis of Salsa20/8 by Tsunoo et al. [1] in 2007 did not deliver any advantage over 256-bit brute force attack, but the following research by Aumasson et al. [2] reduced time complexity to break it from  $2^{255}$  to  $2^{251}$  with 50% success probability. It was improved by Shi et. al [3] in 2012 to  $2^{250}$ . Although this is not critical yet, better attacks on Salsa20/8 may be developed in the future. NeoScript uses Salsa20 and ChaCha20 to produce the final salt. There are no known successful attacks on non-reduced Salsa20 and ChaCha20 other than exhaustive brute force search. NeoScript replaces SHA-256 with BLAKE2s [4] which is a further development of BLAKE-256 [5], one of 5 NIST SHA-3 contest finalists. Based upon ChaCha20, operates with a lower round count of 10, supports keyed hashing, is native little endian and faster significantly than SHA-256 and even BLAKE-256. It could be interfaced directly to PBKDF2 with no need of HMAC. However PBKDF2 constructs derived keys using blocks. It means a minor change in an input datum, such as nonce increment, may not result in an entirely different derived key. A replacement KDF has been developed to address this issue.

## 2 Proof-of-Stake & Proof-of-Work

The underlying database structure for transactions of Bitcoin and other digital currencies is a decentralized ledger, called the blockchain, which stores the entire transaction history. The name stems from the fact that transactions are bundled into blocks; each block in the blockchain (except for the first i.e. genesis block) references a previous block. Each node participating in the Bitcoin network has its own copy of the blockchain, which is synchronized with other nodes using a peer-to-peer protocol. Any implementation of digital currency must have a way to secure its blockchain against attacks. For example, an attacker may spend some money and then reverse the spending transaction by broadcasting his own version of the blockchain, which does not include this transaction; as security of the blockchain does not rely on a single authority, users have no prior knowledge as to which version of the ledger is valid.

In Bitcoin, the security of the network relies on a proof of work (PoW) algorithm in the form of block mining. Each node that wants to participate in mining is required to solve a computationally difficult problem to

ensure the validity of the newly mined block; solutions are rewarded with bitcoins. The protocol is fair in the sense that a miner with  $p$  fraction of the total computational power can win the reward and create a block with the probability  $p$ . An attacker is required to solve the same tasks as the rest of the Bitcoin network; i.e., an attack on Bitcoin will only be successful if the attacker can bring to bear significant computational resources.

Operation of the Bitcoin protocol is such that security of the network is supported by physically scarce resources:

- specialized hardware needed to run computations
- electricity spent to power the hardware

This makes Bitcoin inefficient from a resource standpoint. To increase their share of rewards, Bitcoin miners are compelled to participate in an arms race to continuously deploy more resources in mining. While this makes the cost of an attack on Bitcoin prohibitively high, the ecological unfriendliness of the Bitcoin protocol has resulted in proposals to build similar systems that are much less resource intensive.

One possible decentralized ledger implementation with security not based on expensive computations relies on proof of stake (PoS) algorithms. The idea behind proof of stake is simple: instead of mining power, the probability to create a block and receive the associated reward is proportional to a users ownership stake in the system. An individual stakeholder who has  $p$  fraction of the total number of coins in circulation creates a new block with  $p$  probability. The rationale behind proof of stake is the following: users with the highest stakes in the system have the most interest to maintain a secure network, as they will suffer the most if the reputation and price of the cryptocurrency would diminish because of the attacks. To mount a successful attack, an outside attacker would need to acquire most of the currency, which would be prohibitively expensive for a popular system.

**Table 2:** Vulnerability of proof of work and proof of stake consensus mechanisms to attack vectors

Attack type	Vulnerability		
	PoW	PoS	Delegated PoS
Short range attack (e.g., bribe)	–	+	–
Long range attack	–	+	+ <sup>3</sup>
Coin age accumulation attack	–	maybe <sup>4</sup>	–
Precomputing attack	–	+	–
Denial of service	+	+	+
Sybil attack	+	+	+
Selfish mining	maybe <sup>5</sup>	–	–

This is one of many reasons why a combination of Proof-of-Stake and Proof-of-Work is implemented in Trezarcoin.

### 3 Advanced checkpointing

Advanced checkpointing is a feature that was originally invented and implemented by Feathercoin and its developer Peter Bushnell. Its purpose is to defend the blockchain of Trezarcoin from 51% Attacks. The advanced checkpointing (ACP) feature will remove the need for changes to client software by publishing a feed of checkpoints, via a central node. Checkpointing is a lesser known part of cryptocurrencies. It's a way

to maintain the integrity of the block chain by recording blocks out of it. In bitcoin, checkpoints are stored in the code of the client. This enables it to verify its checkpoints against the copy of the block chain it has downloaded, ensuring that block chain has not been retroactively rewritten in a 51% attack. This means that as the block chain grows, client software has to be updated to store new checkpoints within its code. This leaves clients that have not been updated for a long time vulnerable to attacks on the block chain. The Trezarcoin team intends to solve this problem by separating the checkpoint record from the client software. The team have created a "master node" and updated its client, which will publish the series of checkpoints for Trezarcoin clients to check against.

## 4 Orbitcoin Super Shield

The difficulty of Trezarcoin for PoS and PoW blocks is defined by OSS (Orbitcoin Super Shield) which uses various inputs to retarget the difficulty. To retarget every block, OSS takes the two averaging windows of 5 and 20 blocks, 0.25 damping and further oscillation limiting.

The idea behind OSS is the protection of the network against multipool-mining and leaving the blockchain on a high difficulty. Leaving a blockchain on a difficulty-trap is a common attack on blockchains, which haven't implemented an intelligent retarget-algorithm. The basic advantage of OSS is that it regulates the difficulty according to a few significant measurements, instead of only through the average of the last 2016 blocks like bitcoin does.

## 5 0% Proof-of-Stake

0% Proof-of-Stake adds an additional layer of security to the Trezarcoin protocol. Basically your Coins are starting to stake after a one day period, this period starts from the moment your incoming transaction was broadcasted, including previous PoS block. It is not possible to generate PoS for inputs that are younger than one day and you can not predict, how much time it will take to generate PoS afterwards. It is actually quite similar to traditional solomining with a GPU:

- it depends on the current difficulty (which in turn depends on the amount and activity of the "competitors" - other miners)
- your mining power (but with PoS your mining power not megahashes of mining hardware but amount of coins-days you accumulate in your wallet)
- time which you spend at mining (but with PoS it is not time of GPU/ASIC work, but time while you running wallet online with "old" coins staking)
- your luck

The staking power, usually called "weight" is calculated in the following way:  $(\text{amount of Coins}) * (\text{Coinage} - 1) = \text{Coinweight}$ . The Coinage is counted for each input transaction individually. As explained above inputs with an age smaller than one day have zero weight, although according to the formula they would have a negative value, but the protocol just accounts all this inputs as 0 weight, and aren't staking at all. For example incoming 25 TZC transaction that is one week old have weight =  $25 * (7-1) = 150$  coin-days, increasing continuously with time until generating a PoS finally. After a successful PoS generation coin age reset to zero (so weight = 0 too) and the process starts from the beginning. Maximum possible weight is reached after 16 days, so maximum weight = input size x 15. It is still not possible to calculate the exact time when an individual input will generate a PoS, because it is a stochastic procedure and heavily dependent on luck. Sometimes very "lucky" inputs may generate PoS blocks only a few hours after reaching minimum age. In case of bad luck the exact opposite, a transaction staking a few days or even weeks before generating a PoS reward is possibly as well. Of course having many inputs staking at the same time reduces the luck factor and the timing of rewards becomes more predictable. The main difference between the Trezarcoin

PoS and other PoS systems is best explained in an example. You have 1000 coins and want to stake them for profit continuously, in traditional PoS you can launch your wallet, say, once a month, synchronise, generate a single big stake quickly and shut down the wallet until the next month. You are not motivated to keep it online as much as possible and support the network. In case of TZC, doing the same means losing most of the potential profit. First of all, you want to split these 1000 TZC into smaller parts like 20 to 50 TZC each, so they can stake independently (a process your wallet will take care of automatically, but it is still the fastest way to do it yourself). Second, you want them to stake as soon as possible, and the best way is to keep your wallet online as much as possible. You produce a number of PoS blocks in the process rather than a single one, also you constantly replay blocks, transactions, messages and in short help securing and maintaining the network.

## 6 Conclusion

The primarily advantages and features of Trezarcoin compared to standard Script Coins have been described and evaluated without going too much into details, so even an beginner of the vast topic of cryptocurrencies can understand them. For more details the interested reader should be advised to refer to source code of Trezarcoin.

## 7 References

1. Yukiyasu Tsunoo, Teruo Saito, Hiroyasu Kubo, Tomoyasu Suzaki and Hiroki Nakashima. Differential Cryptanalysis of Salsa20/8, January 2007
2. Jean-Philippe Aumasson, Simon Fischer, Shahram Khazaei, Willi Meier and Christian Rechberger. New Features of Latin Dances: Analysis of Salsa, ChaCha, and Rumba, December of 2007
3. Zhenqing Shi, Bin Zhang, Dengguo Feng and Wenling Wu. Improved Key Recovery Attacks on Reduced-Round Salsa20 and ChaCha, November 2012
4. Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O’Hearn and Christian Winnerlein. BLAKE2: simpler, smaller, fast as MD5, January 2013.
5. Jean-Philippe Aumasson, Luca Henzen, Willi Meier and Raphael C.-W. Phan. SHA-3 proposal BLAKE. Submission to NIST (Round 1/2), 2008.