WhitePaper

Coin of Your Dreams

Desire digital currency

The constantly growing capitalization

Instant transactions

Support Desire 24/7

dev@desire-crypto.com

Contact us if you have any questions

More info on our web desire-crypto.com

Table of Contents

ABSTRACT

Desire is a new P2P digital currency designed to unite all interested users for rewards, exchange and transfer of money to any point of the world, by-passing centralized payment systems.

With instant transactions, anonymity is an undeniable advantage of this currency. To ensure your anonymity, mixing technology is used which makes it impossible to track the sending and receiving destinations.

Desire combines the best features of both Proof-of-Work mining. Its central innovation developed by Ghostlander, is by far the most advanced implementation of this technology to date. Desire supports multiple algorithms to achieve enhanced security. With these features, Desire is targeting users who place a high value on security.

Innovative and Open Development Process

The underlying architecture of the Desire Project is continuously subjected to multiple stages of validation. Through the validation process only the most valuable changes are adopted and integrated. Even an optimal system has limits. Guarantees of reversibility and the highest level of quality are needed. One of Desire's strengths is the ability to adapt and consistently meet the needs of a changing global environment. Building and deconstructing simultaneously is necessary to create the best possible product given the current limitations, and at the same time provide for the ability to further refine the product. In doing so we must accept the possibility that the existing technology has room to improve or that the underlying factors of previous developments have changed. Our methodology allows for continuous evolution of the technology within a changing environment. The resulting development will produce the best possible product at any given time.

Current Technology

NeoScrypt

Strong Memory Intensive Key Derivation Function

INTRODUCTION

Password based key derivation function (KDF) is a deterministic algorithm used to derive a cryptographic key from an input datum known as a password. An additional input datum known as a salt may be employed in order to increase strength of the algorithm against attacks using pre-computed hashes also known as rainbow tables. The derived key length may be specified usually, and one of the most popular uses of KDFs is key stretching. It increases effective length of a user password by constructing an enhanced key to provide with a better resistance against brute force attacks. Another popular use is password storage. Keeping user passwords in unencrypted form is very undesired as it may be possible for an attacker to gain access to the password file and retrieve the passwords stored immediately. Brute force attacks may be the only possible approach against strong KDFs. This kind of attack can be parallelised usually to a great extent. High requirements on computational resources such as processor time and memory space allow to reduce parallelisation efficiency and keep these attacks expensive far beyond reasonable limits.ment

As the name suggests, NeoScrypt is a further development of Scrypt as described in Percival. It is aimed at increased security and better performance on general purpose computer hardware while maintaining comparable costs and requirements. This document focuses on functional differences between NeoScrypt and Scrypt.

SCRYPT SPECIFICATIONS

The most popular implementation of Scrypt employed by many cryptocurrencies since 2011 is N = 1024, r = 1, p = 1 abbreviated usually to (1024, 1, 1) N is the primary parameter defining number of memory segments used and must be a power of 2. May be also described through Nfactor.

N = (1 << (Nfactor + 1))

Nfactor = lb(N) - 1

The default memory segment size for the 32-bit implementation is 128 bytes. r is the segment size multiplier. p is the computational multiplier. They may be also described through rfactor and pfactor respectively.

r = (1 << rfactor)

p = (1 << pfactor)

A single instance of Scrypt utilises (N + 2) * r * 128 bytes of memory space, i.e. 128.25Kb for the (1024, 1, 1) configuration. Actual data mixing in memory is performed by Salsa20, a stream cipher introduced by Bernstein. A reduced strength 8-round implementation has been chosen (Salsa20/8). Every run of the Scrypt core engine executes it 4 * r * N times, i.e. 4096 times for the (1024, 1, 1) configuration. Every execution of Salsa20 mixes one half of a memory segment with itself. The Scrypt core engine has no provisions for key stretching or compressing as well as salting, therefore additional cryptographic functions need to be deployed. In case of crypto-currencies, a typical configuration operates with 80 bytes of input data (block header) which is also a salt. It is passed to PBKDF2, a password based KDF [3] capable of deriving variable length keys with salting. It works with SHA-256, a cryptographic hash function delivering digests up to 32 bytes in size through 64 internal rounds. It doesn't support keyed hashing, therefore a pseudorandom function (PRF) such as HMAC [4] is required, and the whole big endian construction may be called PB-KDF2-HMAC-SHA256. It feeds r * 128 bytes of derived data to the Scrypt core and receives it back after mixing to be used as a salt for another PB-KDF2-HMAC-SHA256 run which compresses 80 bytes of input data into 32 bytes of hash.

CONCLUSIONS

The primary functionality of NeoScrypt and Scrypt has been described and evaluated briefly without much mathematical detail to a cryptography amateur. Certain disadvantages of Scrypt have been outlined. Please refer to the source code and the original Scrypt documentation for additional information should you need any.

MASTERNODES

One of the struggles with the Bitcoin public ledger known as ‘The Blockchain’ is that coins aren't fungible. Once a coin hits the network it's entire history can be traced forward or backwards. For the sake of the discussion we describe Bitcoin as a single tier network (much like a wired network with no vlans) and we describe Desire as a multi-tier network. This is an important distinction. The base of this multi-tiered approach lies in the hands of network clients called Masternodes. In the crypto-currency world we consider a network client a computer with the wallet software running. In terms of Bitcoin all clients of the network are considered equal. The more nodes (or clients) up and available to communicate the stronger the mesh of peer to peer connections is for broadcasting transactions, mining blocks, and coming to a consensus on the ledger.

The power of Desire comes from a rudimentary adjustment to this basic premise of all clients on the network that created equally. Inside the Desire network clients that have/had a single 1000 coin deposit can then in turn attach themselves to another node on the network forming a masternode bond between their local wallet (with the 1000 coins) and the node on the network with no coins but a full copy of the Desire software running and responding to clients on the Desire network. Just like the server hosting this webpage is running software that allows your browser client to talk to it and serve it webpages these Masternodes perform services to support and strengthen the Desire network. The reason for developing this Masternode connection between a local node and a remote node is done for security and reliability. This allows the 1000 coin deposit to remain locked in place and secure. It doesn't have to remain online and accessible it can be safely put away until needed. It also allows the computer responding to client requests on the Desire network to be at a high bandwidth facility. This makes Masternodes highly available and extremely accessible to Desire network clients. Once a wallet has been loaded with a 1000 Desire and started as a Masternode, as long as it remains healthy and responsive to the network for a set period of monitoring it will eventually get entered into the main Masternode list. This is a list of nodes that have all passed this Proof of Service test and that are considered healthy for the network to rely on to sign blocks, relay messages and provide tier two services like fungibility protection. At this point there is two proof of concept offerings for Masternodes that have passed this Proof of Service test. These are InstantX and Darksend or what some like to simply call Mix.

PRIVATSEND

PrivateSend is the feature that gives Desire users full privacy when they use it. It is an improved and extended version of the CoinJoin. In addition to the core concept of CoinJoin, we employ a series of improvements such as decentralization, strong anonymity by using a chaining approach , denominations and passive ahead-of-time mixing. By having a decentralized mixing service within the currency, we gain the ability to keep the currency itself perfectly fungible. At the same time, any user is able to act as an auditor to guarantee the financial integrity of the public ledger without compromising anothers' privacy. PrivateSend uses the fact that a transaction can be formed by multiple parties and made out to multiple parties to merge funds together in a way where they can't be uncoupled thereafter. Given that all PrivateSend transactions are setup for users to pay themselves, the system is highly secure against theft and users coins always remain safe. Currently to mix using PrivateSend requires at least 3 participants. To improve the privacy of the system as a whole we propose using common denominations of 0.1 Desire, 1 Desire, 10 Desire and 100 Desire. In each mixing session, all users should submit the same denominations as inputs and outputs. In addition to denominations, fees should be removed from the transactions and charged in bulk in separate, sporadic unlinkable transactions. PrivateSend is limited to 1000 Desire per session and requires multiple sessions to thoroughly anonymize significant amounts of money. To make the user experience easy and make timing attacks very difficult, PrivateSend runs in a passive mode.

At set intervals, a users client will request to join with other clients via a Masternode. Upon entry into the Masternode, a queue object is propagated throughout the network detailing the denominations the user is looking to anonymize, but no information that can be used to identify the user. Each PrivateSend session can be thought of as an independent event increasing the anonymity of users funds. However each session is limited to three clients, so an observer has a one in three chance of being able to follow a transaction. To increase the quality of anonymity provided, a chaining approach is employed, which funds are sent through multiple Masternodes, one after another.

INSTANTX

By utilizing Masternode quorums, users are able to send and receive instant irreversible transactions. Once a quorum has been formed, the inputs of the transaction are locked to only be spendable in a specific transaction, a transaction lock currently takes about 4 seconds to be set on the network. If a consensus is reached on a lock by the Masternode network, all conflicting transactions or conflicting blocks would be rejected thereafter, unless they matched the exact transaction ID of the lock in place. This will allow vendors to use mobile devices in place of traditional POS systems for real world commerce and users to quickly settle non-commercial transactions face-to-face as with traditional cash. With Desire, this is done without a central authority.

FUTURE TECHNOLOGY

When you invest in DSR you become a stakeholder in a fully open source project aimed at developing the most cutting edge technology for the evolution of cryptocurrency. Investors who choose Desire will be a part of this journey. Current technology is a good starting point however, creating the cryptocurrency of the future requires going beyond the limits of existing technology. The strength of Desire resides in the contract that guarantees investors full participation in the direction of the project, including all future developments and technologies. DSR's potential growth is guaranteed because tomorrow's DSR will not be just another coin, but rather an adaptation to future societal and economic needs. Investors can be confident that when they purchase DSR, they are investing in the present and future growth of a unique cryptocurrency ecosystem. If an algorithm emerges that is more efficient and superior to NeoScrypt, Desire's development team is prepared to adapt DSR's underlying structure to maintain a competitive advantage in the market. As Desire grows the block time can be reduced and the block size can be increased. The adaptability of DSR allows for the blockchain to be maintained while the algorithm is rewritten from scratch. In effect Desire is a blank paper for writing revolutionary code. Writing the code of the future on an already active codebase results in a deployed structure that neither obstructs, limits, nor hinders future development. DSR's architecture allows a subsequent structure to intersect, connect, flank or replace the existing structure. Desire provides a better solution than existing currencies can and guarantees the community full control.