The X22i Algorithm, a Proof of Work Evolution for the Blockchain

Author: Pallas, October 2018

The Problem

A constant concern of the cryptocurrency community is centralization. Bitcoin was born to make people's money free of banks and financial institutions. Therefore, decentralization is a hot word for very valid reasons in the crypto space.

If we consider this issue related to the coin mining scene, it appears clear that there is a need to fight this tendency: ASIC chip makers and FPGA coders are just two examples of "concentration points" of the current mining landscape.

FPGAs and ASICs are expensive machines and can't be used for other tasks (unlike GPUs and CPUs), furthermore FPGA programming is much more complex and resource heavy. So those two technologies lead to centralisation and risk of control by single entities. Instead, everyone should be able to mine their own coins. This is the original bitcoin idea: "One cpu one vote."

The current efforts by the developers of cryptocurrencies haven't been very successful in achieving decentralization. Most coins with enough trade volume are being mined by ASIC or FPGA types of equipment just months after they are out, sometimes even weeks.

The X22i Proposition

The purpose of this white-paper is designing a Proof of Work (PoW) algorithm that can provide the best possible combination of the following points:

1. Make ASIC and FPGA design much more difficult and expensive
2. Allow GPU optimised miners to be developed quickly
3. Allow GPU miners to have maximum efficiency
4. Add quantum resistance
5. Use components which are proven, industry standard algorithms, like sha-2 and sha-3, for best security

FPGA and ASIC Resistance

X22i pursues the goal of ASIC and FPGA resistance by implementing multiple additional features over the standard proof of work algorithm chains like X11. One of these is rising the memory requirements four times (see the SWIFFTX input size in the chart below), which is not a problem for CPU and GPU but much harder for FPGA and ASIC, as they need to either use commodity RAM (giving them no advantage over CPU and GPU) or implement more internal ram, increasing the chip space needed.

Another advantage over the classic PoW algorithms is a much longer algorithm chain: 22 algorithms create the need for a lot of chip space to implement the whole chain, which is very cost-ineffectivefor FPGA and ASIC.

Finally, the bigger plan evolving around X22i is to increase the chain size with further hashing stages (x33i, x24i, etc) to be released periodically. This approach forces the chip designers to revise the design often, meaning more costs and less time for actually using the chip for mining. Moreover, making the chain progressively longer addresses the concern of future FPGA chips bigger in size, and being possibly able to fit the whole X22i chain in a single chip.

GPU Miner Software Development

Being X22i a chain of well known hashing functions, coding a GPU miner is mostly a work of assembling open-source code. At the time this paper was written, there were no GPU implementations of SWIFFTX, but tests made by the author show that the open-source code available, which was written for CPUs, can be easily and quickly adapted to super-parallel GPU code with very good efficiency.

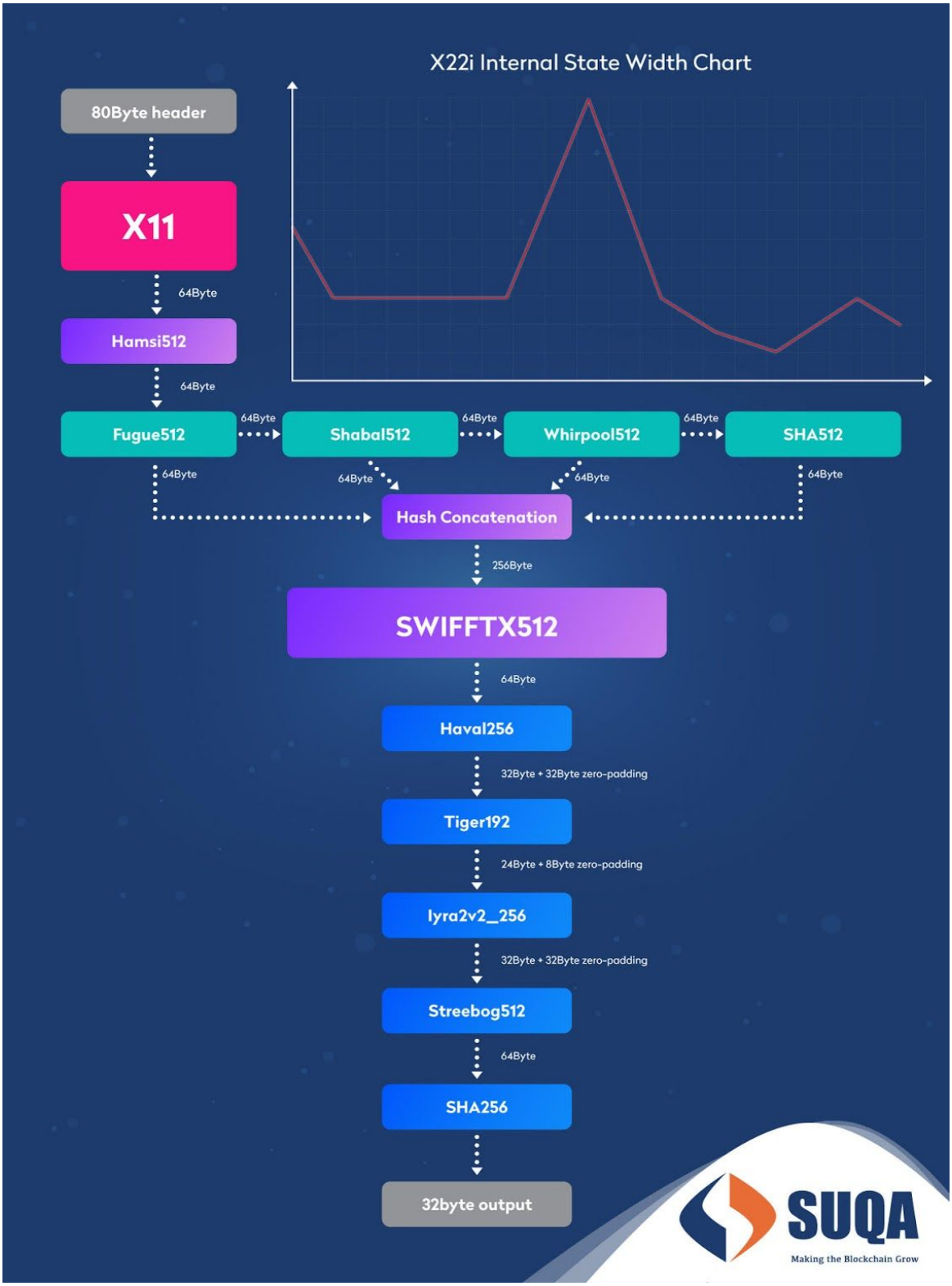
Quantum Resistance

A big concern in the crypto community, linked to centralization but with even worse consequences, if ever exploited, is the possibility of “breaking” the hashing algorithms used in the current coins with a quantum computer. A certain entity with access to this kind of hardware could be able to achieve a huge efficiency advantage over the rest of the miners, or even being able to make an “extreme 51% attack”, reverting a big chunk of the chain and introducing the possibility of double spending, and total control of the blockchain.

To address this issue, X22i introduces a post-quantum element in the chain, SWIFFTX, with lattice-based cryptography.

“Its main attractive features, among others (including no known quantum attack at the time this paper is written) are probably rigorous asymptotic security analyses and asymptotic efficiency.” (https://eprint.iacr.org/2012/343.pdf)

X22i Chart


X22i Internal State Width Chart
80Byte header
X11
64Byte
Hamsi512
64Byte
Fugue512
64Byte
Shabal512
64Byte
Whirpool512
64Byte
SHA512
64Byte
64Byte
64Byte
64Byte
Hash Concatenation
256Byte
SWIFFTX512
64Byte
Haval256
32Byte + 32Byte zero-padding
Tiger192
24Byte + 8Byte zero-padding
lyra2v2_256
32Byte + 32Byte zero-padding
Streebog512
64Byte
SHA256
32byte output
SUQA
Making the Blockchain Grow