QUANTOS NETWORK

PROOFING, HASHING AND VALIDATING CONTENT WITH PROOF OF DNA

WRITEN BY: NICOLAS CLOUTIERS
VERSION 1.0

MARCH 22 2022

GITHUB: QUANTOSNETWORK
EMAIL: DEV@QUANTOS.NETWORK

1.0 Proof of DNA (POD)

Quantos relies on a concept called proof of dna to secure all content such as images, videos, texts and guarantee they are valid.

The advantage over traditional proofing is that it's simple, fast and randomly secure. It was invented by Nicolas Cloutier back in 2018 while he was experimenting with genetic programming.

1.1 Use cases for POD

POD brings a solution to security and verification issues encountered in industries that requires a future proof, high level of security and that manages user content.

Such as: Social Networking, Medical data, Paid content, news outlets, and so on.

1.2 How it works ?

Proof of DNA joins together a few well known concepts:

- Genetic programming
- Natural selection
- Mutation rules of organisms
- Breeding / Survival of the fitest
- The infinite monkey theorem
- The markov algorithm for random content generation

1.2.1 Algorithm explained

POD is using many variables and algorithm which will be explained in details in this section.

1.2.1.1 Initializing Genesis variables

The first step of the algorithm is to define our parameters. For the purpose of explaining we will define them with variables:

Oy = organisms with one or more dna (Ox, Oy)

p = genesis population

F = fitness of each O

sO = Selected organism with higher chance to breed (fitest)

Gx = next generation (x will be replaced by 1,2,3...)

g = gene

m = mutation

T = target

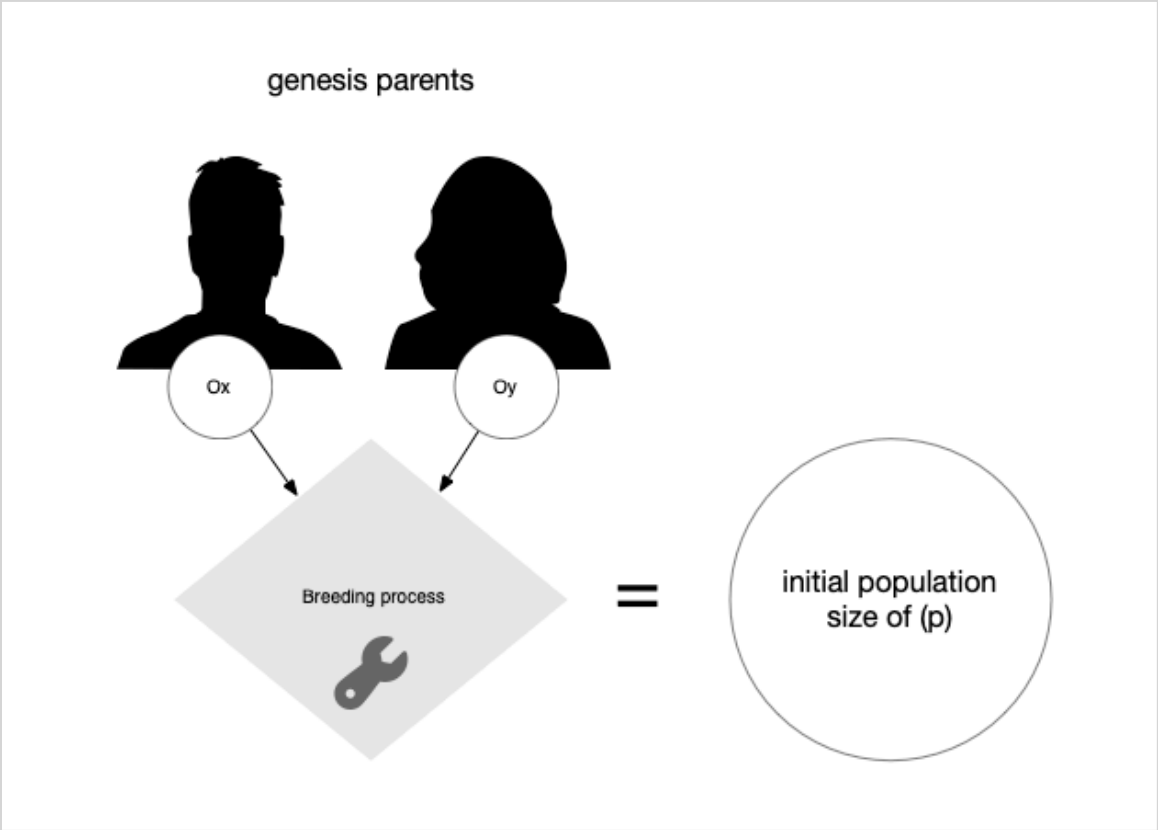


fig 1

1.2.1.2 Generation of genesis data

a) Genesis Parents (Ox and Oy)

To be able to initialize the algorithm we need to find the right parameters for Ox, Oy and p. Ox and Oy can be any result of an abstract breeding process (described in fig 3) these will be our genesis parents the ancestors of everyone could be monkeys, adam and eve pick the names depending on your personal beliefs.

Once we have our Ox and Oy, they need to go trough a breeding process to give birth to our first gene pool (initial population). The size of the population is defined by the variable (p).

Markov Algorithm

a **Markov algorithm** is a string rewriting system that uses grammar-like rules to operate on strings of symbols. Markov algorithms have been shown to be Turing-complete, which means that they are suitable as a general model of computation and can represent any mathematical expression from its simple notation. Markov algorithms are named after the mathematician Andrey Markov Jr.

b) Preparing for the breeding, selecting a target

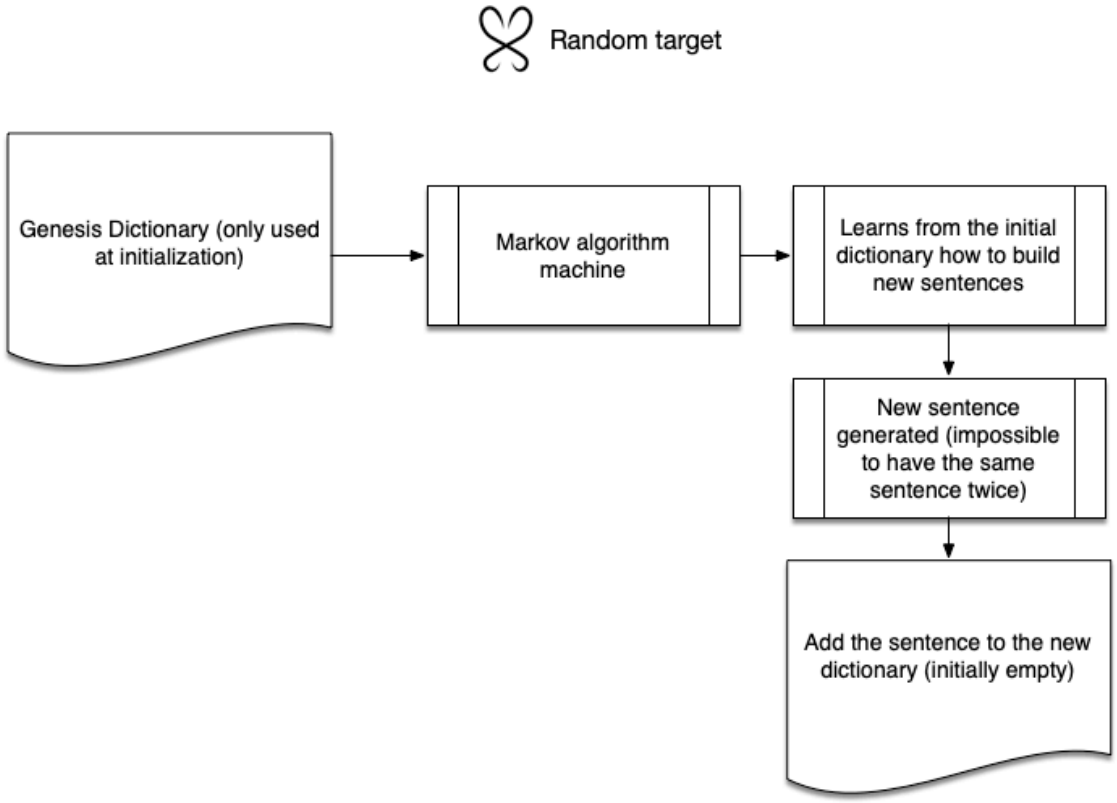


fig 2

For our example, we did select as a source of content to feed our randomizer (markov algorithm) which is a dictionary of 5421 famous english quotes initially. This will define our genesis target. IMPORTANT: the genesis dictionary is only used once to create the rules. A new dictionary is created at each steps from the newly randomly generated targets. Now that we know the value of (T) our target, we can now start breeding from our first population (p)

Breeding and natural selection process

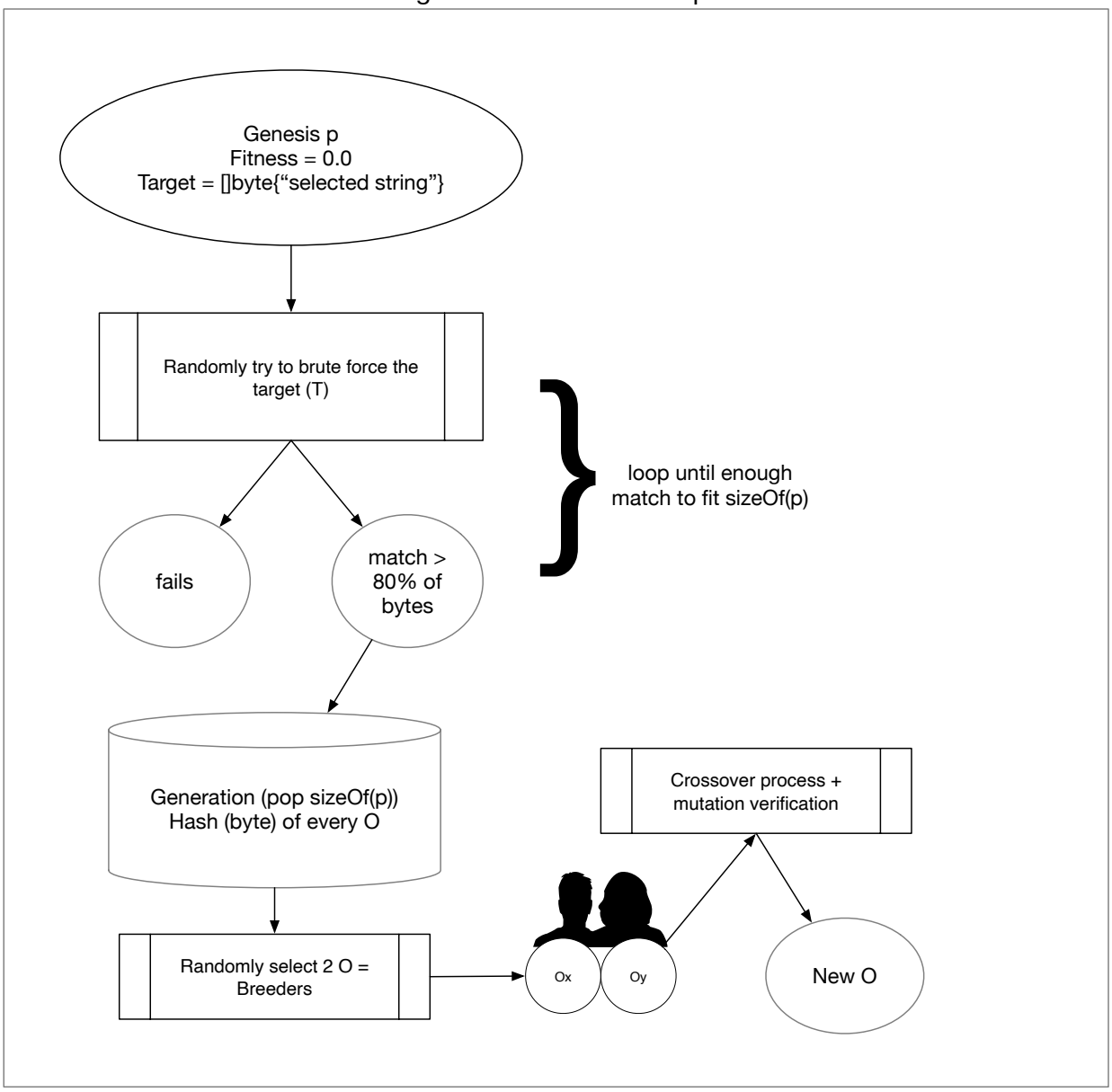


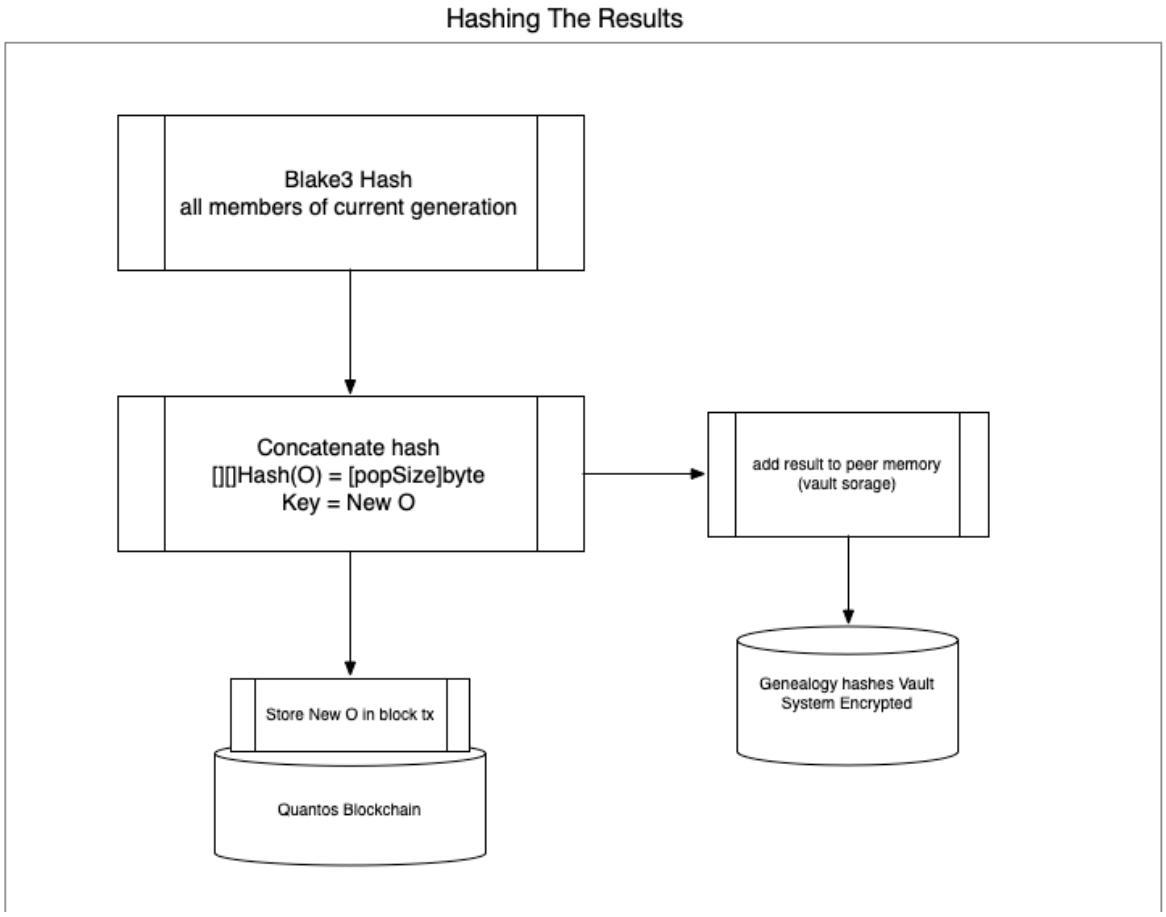
fig 3

c) Breeding and natural selection

The genesis set is selected without great difficulty and in milliseconds. So we can create an initial new dictionary that will make a brute force dictionary attack impossible. Each node will have a different copy of the dictionary which makes it even harder. The dictionary is not used to verify anything and is only a pure source of cryptographically safe randomness.

d) Hashing and Proofing

Once we have the New O, which is the offspring of the process in fig3, we need to setup the proofing and hashing.



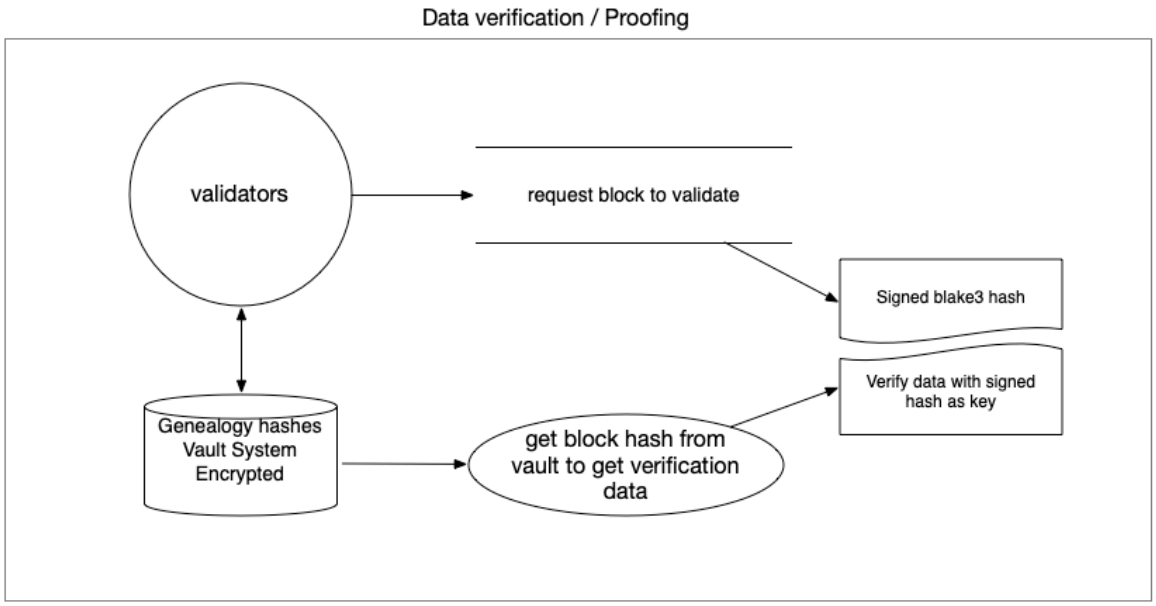


fig 5