Introduction to Randomness

Randomness could be defined as the unpredictability of events considering the information previously available. A cast of dice (lanzamiento de dados) is the much famous example of randomness.

This concept also can be see as a lack of information. Randomness disappears when you have enough knowledge about a system and you are able to predict its next step.

Randomness in Computer Science.

Knowing that randomness is associated with unpredictability, how do computers cope with randomness? Considering the fact that they are deterministic machines that compute code without ambiguity, they should not be able to produce something unpredictable.

A computer is a deterministic machine that should always output the same result for a given input. This is why true randomness needs to be created by something other than basic computing.

Entropy is used to this, entropy is to be viewed as a loss of information about the system. More entropy implies less information which means more uncertainty.

Among the techniques used, we can mention radioactive decay, user activity, atmospheric noise, and so on. This is a way to 'catch' entropy and use

it for the generation of a random number.

Pseudo Randomness

It is a phenomenon that seemingly produces randomness but is completely repeatable.
To circumvent the fact that computing cannot produce true randomness on its own, computers and programs generally use a seed. The seed act as a generator influencing every expected random choice. And although it will seem random, you can replicate the results of every code/experiment if you have the same seed.

Randomness for Cryptography

Randomness in cryptography is used for key generation or transactions.

The ECDSA algorithm is used to sign a transaction using three things provided by the user:

1- Her private key.

2- Transaction/message hash

3- A random number.

The random number should differ between two transactions, otherwise, anyone can compute the private key.

Bitcoin uses the mathematics of elliptic curves digital signatures algorithm (ECDSA).

The use of a deterministic number rather than a random one is now the current method to avoid these vulnerabilities.

To alleviate this problem, secure wallets now change public key addresses after every transaction to avoid this kind of attack among others.

Randomness in Blockchain

Finding a suitable seed for a ‘good’ source of randomness is problematic for two reasons:

1- if the seed is set before the involvement of participants (casting the dice, drawing the cards, etc.), then, the code and the seed will be visible to everyone (Everyone can thus predict the results).

2- if the seed is set after the involvement of participants, you risk choosing a manipulable seed.

So, what source of randomness will you choose?

Data is added to the blockchain through miners, at the arrival of new blocks. Depending on what information you are using from a block, a miner can influence it. And if you draw your seed from a third party, such as an oracle, your trust is based on their reliability.

Blockchains using randomness in their protocols

Several blockchains use randomness to determine the role of agents participating in their protocol. Most notorious examples is Ethereum. They use randomness to determine the roles of validators, such as who is allowed to propose the next block.

The VRF is a function that takes as input a secret key and a nonce and outputs a pseudo-random value. The output from which derives the secret key is then used as a seed for randomness.

The perk of VRF is the Verifiable part, once you have your output, you can prove that this output was correctly computed and hasn’t been tampered with.

Ethereum uses a different mechanism called RANDAO. To produce a random seed, it uses the randomness of every participant such that at least one honest participant ensures randomness.