The Mathematical Foundation of the LeVeL Payment System

The LeVeL protocol offers a robust, secure, private payment experience where payor and payee may withhold their identity from each other and from the community at large, even from the mint that issued the coin they have come to exchange.

This privacy is secured through the mathematical strategy dubbed as "Algorithmic Mutation" -- a step further from the strategy used by other crypto money solutions which may be referred to as "Algorithmic stagnation". Both strategies rely on the notion of one-way-function, OWF (a function which is easy to compute one way but hard to compute in reverse). Virtually all crypto coins rely on a single well-known, well researched OWF: the Elliptic Curve. This 'curve' became the celebrated target for the best hackers in the deepest basements in ministries of cyber warfare around the world. If the Elliptic Curve was not cracked by now, it will surely surrender to higher math very soon, and, at any rate, the fast approaching quantum computers will void the benefit offered by the elliptic curve and evaporate all the monetary wealth carried by the supported currencies.

The LeVeL solution, by contrast, relies on the fact that there is an infinite supply of at-will robust OWFs, *and a 'slow computer' can post them faster than a 'fast computer' can crack them.*

We elaborate. Let's call the ordinary computer (fast as they really are) -- 'slow computers', because indeed they are much slower than the emerging quantum computers, which we will regard as 'fast computers'.

All prevailing crypto currencies rely on the premise that the one-way-function (OWF) they use is robust against attack by 'slow computers'. Yet, there is a wide consensus in the cryptographic community that 'fast computers' will overcome the stagnated OWF used by the prevailing currencies. The answer posted by those currencies is to use a more complex OWF, and hope that fast computers will fail to crack it. It is an unfounded hope since the public is blind about what comes down the pike in terms of fast computation.

The LeVeL team opted for a novel defensive idea based on the following premise: *a slow computer can carve out a new OWF faster -- much faster -- than a fast computer can crack one.* The slow computer does not go head-to-head with the fast computer, but rather hand-to-head: the slow computer is handing over to the fast computer more and more one-way functions to crack. The pace in which the slow computer challenges the fast computer is sufficiently high to keep the fast computer trailing and failing.

It is like guarding a treasure box from a lock-picker by adding more locks to the box than the lock-picker can pick and crack. Suppose a fast computer can crack a typical one-way function within 24 hours. Then the LeVeL coin owner will add a new key to be cracked, say, every 12 hours, and soon enough there will be an ever-growing list of keys to crack and the fast computer will stay hopelessly behind.

The LeVeL coin is designed to easily add locks on the money (more OWF to crack). In fact, every time the LeVeL coin changes hands, the payee retains all the existing locks and adds her new lock. Therefore, *the more a coin trades, the more secure it is*. And since the LeVeL protocol hides the identities of both payor and payee, it is possible for a coin owner to pass the coin to herself often enough, and keep adding keys to the coin. So as fast computers (quantum computers) show up and steadily increase their computational speed, so the LeVeL traders increase the frequency of lock-adding to win the day. This adaptability is the basis of the very strong claim made by LeVeL: being quantum safe.

In the demonstration software submitted for the G20 TechSprint competition we rely on the inherently one-way function: hashing. A hash algorithm accepts a large bit string L as input and generates a small bit string S. This process is irreversible, namely it is mathematically impossible to specify an algorithm that will generate L from S. This is because there are many bit strings L', L", L'", ... etc. that would hash to the same S. One could, though, find some string, say L', that would hash to S. There is always a 'brute force' method to find a large string to hash into a given small string S, simply by trying one large string after the other. This brute force trying is the foundation of the bitcoin mining process. The bitcoin uses the SHA-256 hash function which is considered uncracked, namely it does not allow for a shortcut to find a reverse-hash and a brute force is required. This hashing process is extremely energy intensive: mining a single bitcoin burns energy sufficient to support 50 days of power consumption in an average home in the West. By contrast, the LeVeL protocol, which is an eco-friendly solution, is using the same hash function for its locks, building an insurmountable computational barrier holding up against quantum computers of even a surprise power: simply by adding more and more hashing tasks before the hacker.

The difficulty presented to the brute force attacker is dependent on the bit size l of L and the bit size s of S. The chance for a random l size string to hash into a given S is $1/2^{l-s}$. By choosing

the values of *l* and s, the LeVeL traders control how difficult the task. And the way LeVeL is designed -- the trader (or rather the software on his computing device) -- selects the one-way function of his or her choosing, making it as hard as desired.

The defense of the LeVeL against the possibility of surprise mathematical insight that might harm the coin is to combine the hash function with a randomized permutation. This randomized permutation is described by a patent issued to the LeVeL team (US Patent 10,798,065). As a result, an arbitrary string L undergoes randomized permutation with a permutation key, k_1, to generate L_1 which is a permutation of L. L_1 is then hashed into a small string S_1. The introduction of the k_1-based permutation defines a new one-way function for the fast computers to attack.

A stronger challenge is then presented by using another key, k_2, to process L into L_2, and then hash it to S_2, and so on n times. These n strings S_1, S_2, ... S_n may be concatenated to S, which defines a super hard one-way function.

For educational purposes the first one-way function defined in the demonstration is based on MD5 which is a cracked hash function, but the other four are based on SHA-256 with ever increased permutation complexity.

To summarize: the LeVeL quantum defense strategy is based on using ordinary (slow) computers to develop new one-way functions and pile them up as computational burden facing the attacking (fast) computer. The rate of piling up more hurdles exceeds the rate of negotiating these hurdles by even extra fast quantum computers.