

Claims

1. A computer-implemented method for secure determination of a solution, S , to a computational task by a pooled resource or group having a plurality of participants, P , said group operating in a trust-less, or dealer-free, system or manner, the method comprising the following steps to be executed by each participant:

establishing or joining a group, P , with n participants, $P_{1...n}$, wherein $n \geq 2$;

generating an initial private key share, so^n , and initial public key share, pk_1 , and establishing an initial shared public key, pk_0 , with the group;

performing the task and searching for the solution, S , that determines an answer, A_c , to the task using an intermediate private key, r_n^0 , and the shared public key, pk_0 , of the group;

finding and sharing the solution, S , with the group, or receiving the solution, S , from another participant, thus enabling the group to verify the solution to the task;

calculating with the group a verified shared public key, pk_v , using the initial shared public key, pk_0 , and the intermediate private key, r_n^0 , that provided the solution, S ;

constructing a verified secret key share, sv^0 , by summing the initial private key, so^n , and the intermediate private key, r_n^0 , that provided the solution, S ;

collaborating with all of the other participants, or a threshold number of participants, to construct a verified shared secret key, sk_v , for the verified shared public key, pk_v , using verified secret key shares, sv^0 , wherein the verified shared secret key, sk_v , is used as a code to enable the group to operate collectively to unlock or access a resource or stage of a process.

2. A computer-implemented method according to claim 1, further comprising:

receiving from a client, C , or participant, P , the task and a client public key, pk_c , derived from a client's secret key, sk_c , wherein the client's public key is used in the determination of the solution to the task.

3. A computer-implemented method according to any preceding claim, wherein performing the task and searching for the solution, S , includes:

using the intermediate private key, r_n^0 , to create an intermediate public key, R_n^0 , that is added to the initial shared public key, pk_0 , to create a temporary public key, pk_n^R , that is processed to determine whether the intermediate private key, r_n^0 , was a solution that determined the answer, A_c , sought; and

if the answer, A_c , is not determined, incrementing the value of the intermediate private key, r_n^0 , and repeating the process until the solution, S , is determined, and the answer, A_c , is proven.

4. A computer-implemented method according to any preceding claim, wherein the initial shared public key, pk_0 , and the verified shared public key, pk_v , are generated once.

5. A method according to any preceding claim, wherein the task requires at least one group member to find a solution that, when processed, produces a Cryptocurrency address having a specified pattern.

6. A method according to any preceding claim, wherein the task requires the determination of a set pattern, A , and said set pattern originates from a third-party or another participant, and wherein the third-party generated a third-party secret key, sk_c , and provides to the group a corresponding third party public key, pk_c , wherein $pk_c = sk_c \times G$, and G is an elliptic curve generator, and receiving the third party public key, pk_c , and determining the set pattern using the third party public key, pk_c , with an incremental variable, i , such that $pk = pk_c + i \times G$ when the set pattern is determined, sending to the third-party the incremental variable, i , that enabled determination of the solution, such that the third party can verify the solution using their random secret key, sk_c ,

$$sk = sk_c + i,$$

where $pk = sk \times G$ due to the homomorphic properties of elliptic curve point multiplication.

7. A method according to any preceding claim, wherein the shared secret key, sk_v , sk_o , is established amongst the group using Shamir's secret sharing scheme, and wherein a trust-less relationship is established between each participant P_i by generating their own random degree t polynomial $f_i(x)$, and then securely sending $f_i(j)$ to each other participant P_j , each participant summing all the received points $f_1(i) + f_2(i) + \dots + f_n(i)$ to obtain their secret share $s_i = f(i)$, which is the P_i point on the shared polynomial $f(x)$.

8. A method according to any preceding claim, wherein following creation of the initial private key share, s_0^n , the initial public key share, pk_1 , is computed using an elliptic curve generator G , as $b_i s_n \times G$, wherein the interpolation coefficient b_i is:

$$b_i = \prod_{j \in U, j \neq i} \frac{i}{j-i}, \text{ and}$$

the initial public key shares are then broadcast to all participants and the initial shared public key, pk_o , is then calculated as the sum of any $t + 1$ shares according to:

$$pk = \sum_{i=1}^{t+1} b_i s_i \times G$$

9. A method according to any preceding claim, wherein the task is received from a client, C , or participant, P , said task being to find a private key for a Cryptocurrency address having a specified pattern therein,

wherein the intermediate private key, r_n^0 , is randomly generated and an intermediate public key, R_n^0 , is a multiple of the seed number by r_n^0 with an elliptic curve generator, G , and the temporary public key, R_n^0 , is added to the shared public key, pk_o , and processed to produce a Cryptocurrency address.

10. A method according to any preceding claim, wherein upon all participants having the seed number r_v^0 that provided the solution, S , calculating the verified shared public key from

$$pk_v = pk_o + r_v^0 \times G, \text{ and}$$

constructing a verified secret key, sk_v , from verified secret key shares of each participant, calculated from,

$$s_v^n = s_o^n + r_v^0.$$

11. A method according to any preceding claim, the method further comprises sharing with the group, periodically, intermediate results that are demonstrable of the computational resource applied to the task; and/or

wherein the intermediate results correspond to a portion of the answer, A_c , to the task; and/or

wherein the intermediate results have a proportional level of difficulty, and one or more participants assess the computational resource applied by a fellow participant based on the frequency and/or difficulty of intermediate results submitted.

12. A computer readable storage medium comprising computer-executable instructions which, when executed, configure a processor to perform the method of any one of claims 1 to 11.
13. An electronic device comprising: an interface device; one or more processors coupled to the interface device; a memory coupled to the one or more processors, the memory having stored thereon computer executable instructions which, when executed, configure the one or more processors to perform the method of any one of claims 1 to 11.
14. A node of a blockchain network, the node configured to perform the method of any one of claims 1 to 11.
15. A blockchain network having a node according to claim 11.