WHAT IS CLAIMED IS:

1. A method of validating a multi-party consensus over a limited connection, comprising:

using at least one processor of a validating device for:

transmitting a query to a plurality of computing nodes via a unidirectional secure communication channel, the query having a finite number of possible valid answers;

receiving a limited length string computed based on an aggregated response aggregating a plurality of responses each computed for a multi-party consensus answer to the query by each of at least some of the plurality of computing nodes using a respective one of a plurality of secret components;

computing a plurality of locally computed strings each computed based on a respective one of the finite number of possible valid answers using an aggregated secret aggregating the plurality of secret components;

validating the multi-party consensus answer by comparing the received limited length string to each of the plurality of locally computed strings; and

initiating at least one operation according to an outcome of the validation.

2. The method of claim 1, wherein the validating device applies the validation of the multi-party consensus answer for checking blockchain transactions information relating to the validating device; wherein the validating device is a crypto-currency wallet and the validation of the multi-party consensus response is applied for checking a balance of the crypto-currency wallet;
wherein the at least one operation comprising transmitting at least one cryptocurrency transaction based on the multi-party consensus answer after validated.

3. The method of claim 1, wherein the unidirectional secure communication channel is physically tamper resistant thus supporting reliable and secure one-way communication from the validating device to each of the plurality of computing nodes.

4. The method of claim 1, wherein the validating device communicates with the plurality of computing nodes via at least one access node of the plurality of computing nodes which broadcasts messages received from the validating device to the plurality of computing nodes.

5. The method of claim 1, wherein the secure unidirectional communication channel established with at least one of the plurality of computing nodes is secured by encrypting communication

transmitted from the validating device to the at least one computing node using an encryption key of a respective encryption-decryption key pair uniquely associated with the at least one computing node, the respective encryption key is provided to the validating device as a respective limited length string.

6. The method of claim 1, further comprising the secure unidirectional communication channel established with at least one of the plurality of computing nodes is secured by encrypting communication transmitted from the validating device to the at least one computing node using an encryption key of an encryption-decryption key pair uniquely associated with the at least one computing node, the encryption key is provided to the validating device by at least one trusted controller adapted to distribute to the at least one computing node a decryption key of the encryption-decryption key pair uniquely associated with the at least one computing node, the encryption-decryption key pair is produced deterministically using a pseudorandom number generator initialized with a random seed shared in advance with the validating device, using the shared random seed the validating device generates the encryption key of the encryption-decryption key pair and uses it to establish the encrypt communication transmitted to the at least one computing node.

7. The method of claim 6, wherein the at least one trusted controller is at least one of:

further adapted to periodically distribute a new encryption-decryption key pair to the at least one computing node every predefined period of time, the new encryption-decryption key is produced deterministically using the pseudorandom number generator initialized with the random seed shared with the validating device and a time identifier assigned to the new encryption-decryption key, and

further adapted to distribute a respective encryption-decryption key pair to each of multiple computing nodes, the respective encryption-decryption key pair is produced deterministically using the pseudorandom number generator initialized with the random seed shared with the validating device and an index of the respective computing node.

8. The method of claim 6, wherein the at least one trusted controller communicates with the at least one computing node via a unidirectional secure communication channel similar to the unidirectional secure communication channel of the validating device.

9. The method of claim 1, further comprising transmitting a plurality of queries simultaneously to the plurality of computing nodes such that a later query is transmitted before receiving response to an earlier transmitted query.

10. The method of claim 1, further comprising transmitting the query coupled with a unique identifier which is used by the at least some computing nodes to compute the plurality of responses is used by the validating device to verify that the limited length string corresponds to the query.

11. The method of claim 1, further comprising at least one of the plurality of computing nodes communicates with at least another one of the plurality of computing nodes to support computation of a respective one of the plurality of responses.

12. The method of claim 1, wherein the plurality of secret components are transmitted by the validating device to the plurality of computing nodes; further comprising the validating device transmits the plurality of secret components to the plurality of computing nodes once during an initialization sequence and/or to each of the computing nodes with the query and signs the plurality of secret components transmitted to the plurality of computing nodes using a signing key.

13. The method of claim 1, further comprising applying a two-stage query for a complex query having an extremely large number of possible valid answers as follows:

in a first stage of the two-stage query, the validating device transmits the complex query to the plurality of computing nodes and receives in response a first limited length string computed based on a first multi-party answer computed by at least one of the at least some computing nodes for the multi-party consensus answer to the complex query using the respective secret component, wherein the validating device uses the aggregated secret to extract the first multi-party answer, and

in a second stage of the two-stage query, the validating device transmits a second query requesting the at least some computing nodes to confirm the first multi-party answer received for the complex query, the second query having two possible valid answers, namely correct and incorrect.

14. The method of claim 1, wherein the plurality of secret components is a plurality of symmetric hash functions, each of the plurality of computing nodes uses a respective one of the plurality of symmetric hash functions to compute a respective one of a plurality of hash values based on the

multi-party consensus answer to the query, the limited length string is an aggregated hash value computed by aggregating the plurality of hash values computed by the plurality of computing nodes, the validation is done by comparing the limited length string to each of a set of locally generated limited length strings each generated for a respective one of the finite number of possible valid answers using aggregation of the plurality of symmetric hash functions.

15. The method of claim 14, further comprising publishing a plurality of hash values locally computed by the validating device for each of the finite number of possible valid answers in case, based on analysis of the received limited length string, the validating device determines that the multi-party consensus answer is invalid, each of the plurality of locally computed hash values is computed using a respective one of the plurality of symmetric hash functions, at least one of the plurality of computing nodes identifies at least one malicious computing node of the plurality of computing nodes by detecting incompliance between at least one hash value computed by at least one malicious computing node and a respective one of the plurality of locally computed hash values.

16. The method of claim 14, further comprising using a plurality of deterministic signing functions instead of the plurality of symmetric hash functions, each of the plurality of deterministic signing functions has a respective one of a plurality of signing-verifying key pairs, each of the plurality of computing nodes applies a respective one of the plurality of deterministic signing functions to sign a respective one of the plurality of responses using a signing key of the respective signing-verifying key pair.

17. The method of claim 1, wherein the plurality of secret components are a plurality of partial secret components created from a secret value generated for each of the finite number of possible valid answers using at least one secret sharing algorithm, the validating device generates a respective one of a plurality of random strings each generated for the respective possible valid answer and splits each of the plurality of random strings to a respective set of partial strings using the at least one secret sharing algorithm applied according to at least one configuration parameter, the query transmitted to each of the plurality of computing nodes comprises a respective partial string of each set of partial values, each of the at least some computing nodes transmits its respective partial string corresponding to the multi-party consensus answer, the limited length string is a hash value computed based on the aggregated response aggregating the partial strings received from a sufficient number of the plurality of computing nodes, the sufficient number is

defined by the at least one configuration parameter, the validation is done by comparing the limited length string to each of a set of locally generated limited length strings each generated for a respective one of the finite number of possible valid answers using the secret value.

18. The method of claim 1, wherein the plurality of secret components are a plurality of decryption key components created by splitting a decryption key of an encryption- decryption key pair using at least one threshold decryption algorithm applied according to at least one configuration parameter, the query transmitted to the plurality of computing nodes comprises the query and a plurality of encrypt values encrypting a plurality of random strings each generated by the validation device for a respective one of the finite number of possible valid answers, the plurality of encrypt values are produced by encrypting the plurality of random strings using an encryption key of the encryption-decryption key pair, a sufficient number of the plurality of computing nodes engage in a Multi-Party Computation (MPC) and use their respective decryption key components to decrypt an encrypt value of the plurality of encrypt values corresponding to a certain one of the finite number of possible valid answers as expressed in the multi-party consensus answer, the limited length string is a hash value computed for the decrypted encrypt value, the sufficient number is defined by the at least one configuration parameter, the validation is done by comparing the limited length string to each of a set of locally generated limited length strings each generated for a respective one of the random strings generated for the finite number of possible valid answers.

19. The method of claim 1, wherein the plurality of secret components are a plurality of signing key components created from a signing key of a signing-verifying key pair split using at least one threshold signature algorithm applied according to at least one configuration parameter, the at least one threshold signature algorithm is characterized by producing a deterministic output for a given input, the query transmitted to the plurality of computing nodes comprises the query and a plurality of strings each corresponding to a respective one of the finite number of possible valid answers, a sufficient number of computing nodes of the plurality of computing nodes engage in a Multi-Party Computation (MPC) and use their respective signing key components to produce a deterministic signature for one of the plurality of strings corresponding to an agreed one of the finite number of possible valid answers as expressed in the multi-party consensus answer, the limited length string is a hash value computed for the deterministic signature, the sufficient number is defined by the at least one configuration parameter, the validation is done by comparing the limited length string to

each of a set of locally computed limited length strings each computed for respective deterministic signature generated for a respective one of the plurality of strings using the signing key.

20. An apparatus for validating a multi-party consensus over a limited connection, comprising:

a first interface, adapted to transmit messages to a plurality of computing nodes over a unidirectional secure communication channel;

a second interface adapted to receive limited length strings; and

at least one processor coupled to the first and second interfaces, the at least one processor is adapted to execute a code, the code comprising:

code instructions to transmit a query to the plurality of computing nodes, the query having a finite number of possible valid answers,

code instructions to receive a limited length string computed based on an aggregated response aggregating a plurality of responses each computed for a multi-party consensus answer to the query by each of at least some of the plurality of computing nodes using a respective one of a plurality of secret components,

code instructions to compute a plurality of locally computed strings each computed based on a respective one of the finite number of possible valid answers using an aggregated secret aggregating the plurality of secret components,

code instructions to validate the multi-party consensus answer by comparing the received limited length string to each of the plurality of locally computed strings, and

code instructions to initiate at least one operation according to an outcome of the validation.