

WHAT IS CLAIMED IS:

1. A method for recovery of missing or extra data using a bloom filter, comprising:
storing, in a transaction database of a processing server, a plurality of transaction messages,
5 wherein each transaction message includes a structured data set related to a blockchain
transaction including at least a transaction value; generating, by a generation module of the
processing server, a bloom filter of the plurality of transaction messages, wherein the bloom
filter is generated using a predetermined number of hash rounds and has a size of at least double
a count of the plurality of transaction messages stored in the transaction database; generating, by
10 the generation module of the processing server, a recover message, wherein the recover message
includes at least the count of the plurality of transaction messages, the predetermined number of
hash rounds, the size, and the generated bloom filter; electronically transmitting, by a
transmitting device of the processing server, the generated recover message to at least one
consensus node; receiving, by a receiving device of the processing server, a response message
15 from the at least one consensus node, wherein the response message includes at least one
additional transaction message; and executing, by a querying module of the processing server, a
query on the transaction database to insert the at least one additional transaction message.

2. The method of claim 1, further comprising: generating, by a hashing module of
20 the processing server, a Merkle root for the plurality of transaction messages using a transaction
reference associated with the transaction value included in each of the plurality of transaction
messages; receiving, by the receiving device of the processing server, a proposal message from
the at least one consensus node, wherein the proposal message includes at least a proposed
Merkle root; and verifying, by a verification module of the processing server, that the generated
25 Merkle root is not equivalent to the proposed Merkle root included in each received proposal
message prior to electronically transmitting the generated recover message.

3. The method of claim 2, further comprising: generating, by the hashing module of
the processing server, a new Merkle root for the plurality of transaction messages and the at least
30 one additional transaction message using a transaction reference associated with the transaction
value included in each respective transaction message; and verifying, by the verification module

of the processing server, that the new Merkle root is equivalent to the proposed Merkle root included in each received proposal message.

4. The method of claim 2, further comprising: generating, by the hashing module of
5 the processing server, the transaction reference associated with the transaction value included in each of the plurality of transaction messages by hashing the respective transaction value using a predetermined hashing algorithm.

5. The method of claim 2, further comprising: sorting, by the querying module of the
10 processing server, the transaction reference associated with the transaction value included in each of the plurality of transaction messages based on a natural ordering prior to generating the Merkle root.

6. The method of claim 1, wherein each of the plurality of transaction messages
15 further includes a specific slot identifier, the recover message and response message each further includes the specific slot identifier, and the at least one additional transaction message includes the specific slot identifier.

7. The method of claim 1, wherein the bloom filter is generated using a
20 predetermined hashing algorithm for each of the predetermined number of hash rounds.

8. A method for recovery of missing or extra data using a bloom filter, comprising:
storing, in a transaction database of a processing server, a plurality of transaction messages,
wherein each transaction message includes a structured data set related to a blockchain
25 transaction including at least a transaction value; generating, by a generation module of the processing server, a bloom filter of the plurality of transaction messages, wherein the bloom filter is generated using a predetermined number of hash rounds and has a size of at least double a count of the plurality of transaction messages stored in the transaction database; generating, by
the generation module of the processing server, a recover message, wherein the recover message
30 includes at least the count of the plurality of transaction messages, the predetermined number of hash rounds, the size, and the generated bloom filter; electronically transmitting, by a

transmitting device of the processing server, the generated recover message to at least one consensus node; receiving, by a receiving device of the processing server, a response message from the at least one consensus node, wherein the response message includes at least a second bloom filter, an indicated number of hash rounds, an indicated filter size, and a number of expected transaction messages; identifying, by a data identification module of the processing server, at least one transaction message of the plurality of transaction messages not included in the second bloom filter based on the included transaction value and the indicated number of hash rounds, indicated filter size, number of expected transaction messages, and second bloom filter; and executing, by a querying module of the processing server, a query on the transaction database to delete the at least one identified transaction message.

9. The method of claim 8, wherein the number of expected transaction messages is less than the count of the plurality of transaction messages.

10. The method of claim 8, wherein the indicated filter size is at least double the number of expected transaction messages.

11. The method of claim 8, wherein an updated count of the plurality of transaction messages after execution of the query is equivalent to the number of expected transaction messages.

12. The method of claim 8, further comprising: generating, by a hashing module of the processing server, a Merkle root for the plurality of transaction messages using a transaction reference associated with the transaction value included in each of the plurality of transaction messages; receiving, by the receiving device of the processing server, a proposal message from the at least one consensus node, wherein the proposal message includes at least a proposed Merkle root; and verifying, by a verification module of the processing server, that the generated Merkle root is not equivalent to the proposed Merkle root included in each received proposal message prior to electronically transmitting the generated recover message.

13. The method of claim 12, further comprising: generating, by the hashing module of the processing server, a new Merkle root for the plurality of transaction messages after deletion of the at least one identified transaction message using a transaction reference associated with the transaction value included in each respective transaction message; and verifying, by the
5 verification module of the processing server, that the new Merkle root is equivalent to the proposed Merkle root included in each received proposal message.

14. The method of claim 12, further comprising: generating, by the hashing module of the processing server, the transaction reference associated with the transaction value included in
10 each of the plurality of transaction messages by hashing the respective transaction value using a predetermined hashing algorithm.

15. The method of claim 12, further comprising: sorting, by the querying module of the processing server, the transaction reference associated with the transaction value included in
15 each of the plurality of transaction messages based on a natural ordering prior to generating the Merkle root.

16. The method of claim 8, wherein each of the plurality of transaction messages further includes a specific slot identifier, and the recover message and response message each
20 further includes the specific slot identifier.

17. The method of claim 8, wherein the bloom filter is generated using a predetermined hashing algorithm for each of the predetermined number of hash rounds.

25 18. A system for recovery of missing or extra data using a bloom filter, comprising: a processing server; a transaction database of the processing server configured to store a plurality of transaction messages, wherein each transaction message includes a structured data set related to a blockchain transaction including at least a transaction value; a generation module of the processing server configured to generate a bloom filter of the plurality of transaction messages,
30 wherein the bloom filter is generated using a predetermined number of hash rounds and has a size of at least double a count of the plurality of transaction messages stored in the transaction

database, and a recover message, wherein the recover message includes at least the count of the plurality of transaction messages, the predetermined number of hash rounds, the size, and the generated bloom filter; a transmitting device of the processing server configured to electronically transmit the generated recover message to at least one consensus node; a receiving device of the processing server configured to receive a response message from the at least one consensus node, wherein the response message includes at least one additional transaction message; and a querying module of the processing server configured to execute a query on the transaction database to insert the at least one additional transaction message.

19. The system of claim 18, further comprising: a verification module of the processing server; and a hashing module of the processing server configured to generate a Merkle root for the plurality of transaction messages using a transaction reference associated with the transaction value included in each of the plurality of transaction messages, wherein the receiving device of the processing server is further configured to receive a proposal message from the at least one consensus node, wherein the proposal message includes at least a proposed Merkle root, and the verification module of the processing server is configured to verify that the generated Merkle root is not equivalent to the proposed Merkle root included in each received proposal message prior to electronically transmitting the generated recover message.

20. The system of claim 19, wherein the hashing module of the processing server is further configured to generate a new Merkle root for the plurality of transaction messages and the at least one additional transaction message using a transaction reference associated with the transaction value included in each respective transaction message, and the verification module of the processing server is further configured to verify that the new Merkle root is equivalent to the proposed Merkle root included in each received proposal message.

21. The system of claim 19, wherein the hashing module of the processing server is further configured to generate the transaction reference associated with the transaction value included in each of the plurality of transaction messages by hashing the respective transaction value using a predetermined hashing algorithm.

22. The system of claim 19, wherein the querying module of the processing server is further configured to sort the transaction reference associated with the transaction value included in each of the plurality of transaction messages based on a natural ordering prior to generating the Merkle root.

5

23. The system of claim 18, wherein each of the plurality of transaction messages further includes a specific slot identifier, the recover message and response message each further includes the specific slot identifier, and the at least one additional transaction message includes the specific slot identifier.

10

24. The system of claim 18, wherein the bloom filter is generated using a predetermined hashing algorithm for each of the predetermined number of hash rounds.

25. A system for recovery of missing or extra data using a bloom filter, comprising: a
15 processing server; a transaction database of the processing server configured to store a plurality of transaction messages, wherein each transaction message includes a structured data set related to a blockchain transaction including at least a transaction value; a generation module of the processing server configured to generate a bloom filter of the plurality of transaction messages, wherein the bloom filter is generated using a predetermined number of hash rounds and has a
20 size of at least double a count of the plurality of transaction messages stored in the transaction database, and a recover message, wherein the recover message includes at least the count of the plurality of transaction messages, the predetermined number of hash rounds, the size, and the generated bloom filter; a transmitting device of the processing server configured to electronically transmit the generated recover message to at least one consensus node; a receiving device of the
25 processing server configured to receive a response message from the at least one consensus node, wherein the response message includes at least a second bloom filter, an indicated number of hash rounds, an indicated filter size, and a number of expected transaction messages; a data identification module of the processing server configured to identify at least one transaction message of the plurality of transaction messages not included in the second bloom filter based on
30 the included transaction value and the indicated number of hash rounds, indicated filter size, number of expected transaction messages, and second bloom filter; and a querying module of the

processing server configured to execute a query on the transaction database to delete the at least one identified transaction message.

26. The system of claim 25, wherein the number of expected transaction messages is
5 less than the count of the plurality of transaction messages.

27. The system of claim 25, wherein the indicated filter size is at least double the number of expected transaction messages.

10 28. The system of claim 25, wherein an updated count of the plurality of transaction messages after execution of the query is equivalent to the number of expected transaction messages.

29. The system of claim 25, further comprising: a verification module of the
15 processing server; and a hashing module of the processing server configured to generate a Merkle root for the plurality of transaction messages using a transaction reference associated with the transaction value included in each of the plurality of transaction messages, wherein the receiving device of the processing server is further configured to receive a proposal message from the at least one consensus node, wherein the proposal message includes at least a proposed
20 Merkle root, and the verification module of the processing server is configured to verify that the generated Merkle root is not equivalent to the proposed Merkle root included in each received proposal message prior to electronically transmitting the generated recover message.

30. The system of claim 29, wherein the hashing module of the processing server is
25 configured to generate a new Merkle root for the plurality of transaction messages after deletion of the at least one identified transaction message using a transaction reference associated with the transaction value included in each respective transaction message, and the verification module of the processing server is configured to verify that the new Merkle root is equivalent to the proposed Merkle root included in each received proposal message.

30

31. The system of claim 29, wherein the hashing module of the processing server is further configured to generate the transaction reference associated with the transaction value included in each of the plurality of transaction messages by hashing the respective transaction value using a predetermined hashing algorithm.

5

32. The system of claim 29, wherein the querying module of the processing server is further configured to sort the transaction reference associated with the transaction value included in each of the plurality of transaction messages based on a natural ordering prior to generating the Merkle root.

10

33. The system of claim 25, wherein each of the plurality of transaction messages further includes a specific slot identifier, and the recover message and response message each further includes the specific slot identifier.

15

34. The system of claim 25, wherein the bloom filter is generated using a predetermined hashing algorithm for each of the predetermined number of hash rounds.