

WHAT IS CLAIMED IS:

- 1 1. A method comprising:
2 performing, by a node in a plurality of nodes, an identity set generation
3 process;
4 determining, by the node, a leader node;
5 diffusing, by the node, an identity set to the plurality of nodes; and
6 determining, by the node, a majority set including identities occurring in at
7 least one half of the identity sets, wherein the leader node diffuses the majority set of
8 the leader node to the plurality of nodes.
- 1 2. The method of claim 1, wherein determining the leader node further
2 comprises:
3 determining, by the node, a smallest hash value of a plurality of hash
4 values received from the plurality of nodes;
5 determining, by the node, an identity of the identity set associated with the
6 smallest hash value; and
7 selecting, by the node, a second node associated with the smallest hash
8 value to be the leader node.
- 1 3. The method of claim 2, wherein the node is the second node.
- 1 4. The method of claim 1 further comprising:
2 receiving, by the node, the majority set from the leader node;
3 verifying, by the node, the majority set of the leader node; and
4 updating, by the node, the identity set based on the majority set of the
5 leader node.
- 1 5. The method of claim 1, wherein verifying the majority set of the
2 leader node further comprises:
3 determining, by the node, that the majority set includes identities occurring
4 in at least one half of the identity sets.

1 6. The method of claim 1, wherein performing the identity set
2 generation process further comprises:
3 generating, by the node, a public key, a private key, and a random
4 challenge string;
5 transmitting, by the node, the random challenge string to the plurality of
6 nodes;
7 receiving, by the node, a plurality of random challenge strings from the
8 plurality of nodes;
9 generating, by the node, a challenge set comprising the plurality of
10 random challenge strings and the random challenge string;
11 determining, by the node, a nonce that solves a proof of work;
12 transmitting, by the node, a tuple comprising the public key, the nonce, a
13 hash value from the proof of work, and the challenge set to the plurality of nodes;
14 receiving, by the node, a plurality of tuples from the plurality of nodes; and
15 verifying, by the node, the plurality of tuples, wherein if a tuple of the
16 plurality of tuples is valid then storing a public key associated with the tuple in an
17 identity set.

1 7. The method of claim 6, wherein the random challenge string
2 comprises κ bits, and wherein each node of the plurality of nodes generates a different
3 random challenge string.

1 8. The method of claim 6, wherein inputs to the proof of work include
2 the public key, the challenge set, and the nonce.

1 9. The method of claim 8, wherein determining the nonce that solves
2 the proof of work further comprises:
3 determining, by the node, the hash value which is an output of the proof of
4 work; and
5 determining, by the node, that the hash value is less than a predetermined
6 difficulty parameter.

1 10. The method of claim 1 further comprising:
2 splitting, by the node, the identity set into buckets, wherein each bucket is
3 processed in a round.

1 11. A node comprising:
2 a processor;
3 a memory device; and
4 a computer-readable medium coupled to the processor, the computer-
5 readable medium comprising code executable by the processor for implementing a
6 method comprising:
7 performing, by a node in a plurality of nodes, an identity generation set
8 process;
9 determining a leader node;
10 diffusing an identity set to the plurality of nodes; and
11 determining a majority set including identities occurring in at least one half
12 of the identity sets, wherein the leader node diffuses the majority set of the leader node
13 to the plurality of nodes.

1 12. The node of claim 11, wherein determining the leader node further
2 comprises:
3 determining a smallest hash value of a plurality of hash values received
4 from the plurality of nodes;
5 determining an identity of the identity set associated with the smallest
6 hash value; and
7 selecting a second node associated with the smallest hash value to be the
8 leader node.

1 13. The node of claim 12, wherein the node is the second node.

1 14. The node of claim 11, wherein the method further comprises:
2 receiving the majority set from the leader node;
3 verifying the majority set of the leader node; and

4 updating the identity set based on the majority set of the leader node.

1 15. The node of claim 11, wherein verifying the majority set of the
2 leader node further comprises:
3 determining that the majority set includes identities occurring in at least
4 one half of the identity sets.

1 16. The node of claim 11, wherein performing the identity set
2 generation process further comprises:
3 generating a public key, a private key, and a random challenge string;
4 transmitting the random challenge string to the plurality of nodes;
5 receiving, by the node, a plurality of random challenge strings from the
6 plurality of nodes;
7 generating a challenge set comprising the plurality of random challenge
8 strings and the random challenge string;
9 determining a nonce that solves a proof of work;
10 transmitting a tuple comprising the public key, the nonce, a hash value
11 from the proof of work, and the challenge set to the plurality of nodes;
12 receiving a plurality of tuples from the plurality of nodes; and
13 verifying the plurality of tuples, wherein if a tuple of the plurality of tuples is
14 valid then storing a public key associate with the tuple in an identity set.

1 17. The node of claim 16, wherein the random challenge string
2 comprises κ bits, and wherein each node of the plurality of nodes generates a different
3 random challenge string.

1 18. The node of claim 16, wherein inputs to the proof of work include
2 the public key, the challenge set, and the nonce.

1 19. The node of claim 18, wherein determining the nonce that solves
2 the proof of work further comprises:
3 determining the hash value which is an output of the proof of work; and

4 determining that the hash value is less than a predetermined difficulty
5 parameter.

1 20. The node of claim 11, wherein the method further comprises:
2 splitting, by the node, the identity set into buckets, wherein each bucket is
3 processed in a round.