

Claims

1. A computer-implemented method comprising:

storing a previous blockchain transaction in a blockchain maintained by a blockchain network, the previous blockchain transaction including a locking script representing an instance of a parent execution thread;

creating and/or validating a first blockchain transaction that includes a plurality of locking scripts each representing an instance of a child execution thread which is forked from the parent execution thread in response to a corresponding first construct or operation included in the locking script of the previous blockchain transaction; and

communicating the first blockchain transaction on the blockchain network for storage in the blockchain wherein:

the first blockchain transaction points to the locking script of the previous blockchain transaction which represents the parent execution thread.

2. The computer-implemented method of claim 1, wherein

the locking script of the previous blockchain transaction includes a bytecode sequence and optionally interpreter code, and at least one of the plurality of locking scripts of the first blockchain transaction includes a copy of the bytecode sequence and optionally a copy of the interpreter code of the locking script of the previous blockchain transaction, or

the locking script of the previous blockchain transaction includes a bytecode sequence and optionally interpreter code, and the plurality of locking scripts of the first blockchain transaction include i) a copy of the bytecode sequence and optionally a copy of the interpreter code of the locking script of the previous blockchain transaction or ii) a distinct bytecode sequence and optionally interpreter code that belongs to a collection of valid locking scripts.

3. The computer-implemented method of claim 1 or claim 2 , wherein:

the plurality of locking scripts of the first blockchain transaction each include execution state information (such as a next execution pointer) for a respective child execution thread.

4. The computer-implemented method of any of claims 1 to 3, further comprising:

validating the first blockchain transaction by matching at least one of the plurality of locking scripts of the first blockchain transaction to the locking script of the previous blockchain transaction; and/or

validating the first blockchain transaction by matching the bytecode sequence and optionally interpreter code for each one of the plurality of locking scripts of the first blockchain transaction to the bytecode sequence and optionally interpreter code of the locking script of the previous blockchain transaction, whereby validation of the first blockchain transaction is determined based on the results of such matching; and/or

validating the first blockchain transaction by generating a hash data value that corresponds the locking script of the previous blockchain transaction, adding the hash data value to a collection of hash data values that correspond to the collection of valid locking scripts, and comparing a hash data value generated for each locking script of the first blockchain transaction to the collection of hash data values, whereby validation of the first blockchain transaction is determined based on the results of such hash data value comparisons; and/or

validating the first blockchain transaction by checking execution state information of the child execution thread instances representing by the plurality of locking scripts of the first spending blockchain transaction.

5. The computer-implemented method of any of claims 1 to 4, further comprising:

creating and/or validating a second blockchain transaction that includes a locking script representing an instance of a child execution thread joined from a plurality of parent execution thread instances represented by locking scripts of at least one other previous blockchain transaction stored in the blockchain in response to a corresponding

second construct or operation included in at least one locking script of the at least one other previous blockchain transaction; and

communicating the second blockchain transaction on the blockchain network for storage in the blockchain.

6. The computer-implemented method of claim 5, wherein:

the second blockchain transaction points to a plurality of locking scripts of the at least one other previous blockchain transaction, wherein the plurality of locking scripts represent the plurality of parent execution thread instances joined by the second block transaction; and/or wherein:

the plurality of locking scripts of the at least one other previous blockchain transaction each include a bytecode sequence and optionally interpreter code, and the locking script of the second blockchain transaction includes a copy of the bytecode sequence and optionally the interpreter code of one of the locking scripts of the one or more previous blockchain transaction, or

the plurality of locking scripts of the at least one other previous blockchain transactions each include a bytecode sequence and optionally interpreter code, and the locking script of the second blockchain transaction includes i) a copy of the bytecode sequence and optionally the interpreter code of one of the locking scripts of the one or more previous blockchain transaction, or ii) a distinct bytecode sequence and optionally interpreter code that belongs to a collection of valid locking scripts.

7. The computer-implemented method of claim 5 to 6 , wherein:

the locking script of the second blockchain transaction includes execution state information for the child execution thread joined from the plurality of parent execution thread instances.

8. The computer-implemented method of any of claims 5 to 7, further comprising:

validating the second blockchain transaction by matching a locking script of the at least one other previous blockchain transaction to the locking script of the second blockchain transaction; and/or

validating the second blockchain transaction by matching the bytecode sequence and optionally interpreter code of a locking script of the at least one other previous blockchain transaction to the bytecode sequence and optionally interpreter code of the locking script of the second blockchain transaction, whereby validation of the second blockchain transaction is determined based on the results of such matching; and/or

validating the second blockchain transaction by generating a hash data value that corresponds a locking script of the at least one other previous blockchain transaction, adding the hash data value to a collection of hash data values that correspond to the collection of valid locking scripts, and comparing a hash data value generated for the locking script of the second blockchain transaction to the collection of hash data values, whereby validation of the second blockchain transaction is determined based on the results of such hash data value comparisons; and/or

validating the second blockchain transaction by checking execution state information of the child execution thread represented by the locking script of the second blockchain transaction.

9. The computer-implemented method of any of claims 1 to 8, further comprising:

creating and/or validating a third blockchain transaction that includes a plurality of locking scripts representing execution threads of at least one other previous blockchain transaction stored in the blockchain for inter-thread communication and control in response to a corresponding third construct or operation included in at least one locking script of the least one other previous blockchain transaction; and

communicating the third blockchain transaction on the blockchain network for storage in the blockchain.

10. The computer-implemented method of claim 9, wherein:

the third blockchain transaction points to a plurality of locking scripts of the at least one other previous blockchain transaction, wherein the plurality of locking scripts represent the plurality of execution threads that are managed or controlled by the third blockchain transaction and/or

wherein:

the plurality of locking scripts of the at least one other previous blockchain transaction each include a bytecode sequence and optionally interpreter code, and the plurality of locking scripts of the third blockchain transaction includes a copy of the bytecode sequence and optionally the interpreter code of one of the locking scripts of the at least one other previous blockchain transaction, or

the plurality of locking scripts of the at least one other previous blockchain transaction each include a bytecode sequence and optionally interpreter code, and the locking scripts of the third blockchain transaction includes i) a copy of the bytecode sequence and optionally the interpreter code of one of the locking scripts of the at least one other previous blockchain transaction, or ii) a distinct bytecode sequence and optionally interpreter code that belongs to a collection of valid locking scripts.

11. The computer-implemented method of claims 9 to 10, wherein:

the locking scripts of the third blockchain transaction include execution state information for the execution threads managed and controlled by the third blockchain transaction.

12. The computer-implemented method of any of claims 9 to 11, further comprising:

validating the third blockchain transaction by matching a locking script of the at least one other previous blockchain transaction to a locking script of the third blockchain transaction; and/or

validating the third blockchain transaction by matching the bytecode sequence and optionally interpreter code of a locking script of the at least one other previous blockchain transaction to the bytecode sequence and optionally interpreter code of a

locking script of the third blockchain transaction, whereby validation of the third blockchain transaction is determined based on the results of such matching; and/or

validating the third blockchain transaction by generating a hash data value that corresponds a locking script of the at least one other previous blockchain transaction, adding the hash data value to a collection of hash data values that correspond to the collection of valid locking scripts, and comparing a hash data value generated for a locking script of the third blockchain transaction to the collection of hash data values, whereby validation of the third blockchain transaction is determined based on the results of such hash data value comparisons; and/or

validating the third blockchain transaction by checking execution state information of the execution threads managed and controlled by the third blockchain transaction.

13. A computer-implemented method according to any one of claims 1 to 12, wherein the first blockchain transaction and/or the second blockchain transaction and/or the third blockchain transaction is part of a smart contract.

14. A system, comprising:

a processor; and

memory including executable instructions that, as a result of execution by the processor, causes the system to perform at least part of the computer-implemented method of any one of claims 1 to 13.

15. A non-transitory computer-readable storage medium having stored thereon executable instructions that, as a result of being executed by a processor of a computer system, cause the computer system to perform at least part of the computer-implemented method of any one of claims 1 to 13.