

CLAIMS

What is claimed is:

1. A tangible, non-transitory, machine-readable medium storing instructions that when executed by one or more processors effectuate operations comprising:
 - receiving, with a computer system, a request to prune data stored in a tamper-evident data store, wherein:
 - the data includes first program state of a first program at a first time,
 - the first program is one of a plurality of programs having program state stored in the tamper-evident data store,
 - the request uniquely identifies the first program with an identifier that distinguishes the first program from a second program among the plurality of programs,
 - the tamper-evident data store comprises a directed acyclic graph of cryptographic hash pointers,
 - a first subset of the cryptographic hash pointers have cryptographic hash values based on the first program state of the first program at the first time,
 - a second subset of the cryptographic hash pointers have cryptographic hash values based on second program state of the first program at a second time,
 - a third subset of the cryptographic hash pointers have cryptographic hash values based on the first program state of the first program at the first time, the second program state of the first program at the second time, and program state of the second program from before the first time, and
 - the tamper-evident data store comprises or is associated with a record that links the first subset of the cryptographic hash pointers to the second subset of the cryptographic hash pointers;
 - in response to the request, based on the identifier and the record that links the first subset of the cryptographic hash pointers to the second subset of the cryptographic hash pointers, with the computer system, selecting:
 - a first subset of nodes of the directed acyclic graph having first content based on the first program state of the first program at the first time, the hash values of first subset of the cryptographic hash pointers being based on the first content, and
 - a second subset of nodes of the directed acyclic graph having second content

based on the first program state of the first program at the first time, the hash values of second subset of the cryptographic hash pointers being based on the second content;

pruning, with the computer system, the first program state and the second program state from the tamper-evident data store by changing the first content and the second content in memory, wherein:

after the pruning, cryptographic hash values of the directed acyclic graph of cryptographic hash pointers are internally consistent, and

after the pruning, program state of second program from before the first time remains in the tamper-evident data store.

2. The medium of claim 1, wherein:

receiving comprises receiving an invocation of a subroutine of the first program sent to an address of the first program on a permissioned decentralized computing platform configured to persist state to the tamper-evident data store;

the address is based on an address of a creator of the first program and a count of programs previously deployed by the creator;

the subroutine, upon being interpreted to bytecode, invokes a pruning opcode of a virtual machine, upon replicated instances of which virtual machine the first program is redundantly executed on a plurality of computing nodes of the decentralized computing platform;

the record comprises a linked list of nodes in the tamper-evident data store based on program state of the first program at different times;

the first content comprises a cryptographic transaction hash of a transaction on the decentralized computing platform by which the first program was deployed;

pruning comprises changing the cryptographic transaction hash of the transaction on the decentralized computing platform by which the first program was deployed;

the second content comprises a cryptographic storage hash based on a root node of first trie of cryptographic hash pointers having program state of the first program but not the second program;

pruning comprises changing the cryptographic storage hash;

the second content comprises a transaction by which the first program was invoked to mutate program state of the first program from the first program state to the second program state;

the transaction is stored in a second trie of cryptographic hash pointers, the second trie of cryptographic hash pointers having transactions with programs among the plurality of programs other than the first program; and

pruning comprises changing, in the second trie of cryptographic hash pointers, the transaction by which the first program was invoked to mutate program state of the first program from the first program state to the second program state.

3. The medium of claim 1, wherein:

pruning comprises de-indexing entries in the tamper-evident data store and designating entries as having been pruned in the tamper-evident data store.

4. The medium of claim 1, wherein pruning comprises:

iterating through each instance of program state of the first program, including the first program state and the second program state and, at each of at least a plurality of iterations:

in a current-iteration entry in a state trie, changing a current-iteration cryptographic code hash value based on code of the first program;

in the current-iteration entry in a state trie, changing a current-iteration cryptographic storage hash value based on data stored by the first program;

changing values other than a current-iteration cryptographic receipt hash in a current-iteration transaction entry identified by a cryptographic transaction hash in the current-iteration entry in the state trie;

changing values in an entry in a receipt trie identified by the current-iteration cryptographic receipt hash; and

accessing a current-iteration pointer to a predecessor entry in a state trie.

5. The medium of claim 1, wherein pruning is executed synchronously between instances in which transaction tries are added to the directed acyclic graph of cryptographic hash pointers.

6. The medium of claim 1, the operations comprising synchronizing a new computing node with a network of computing nodes storing the tamper-evident data store by chronologically

executing a history of transactions stored in the tamper-evident data, wherein synchronizing comprises:

determining that a first transaction has not been pruned, and in response, executing the first transaction; and

determining that a second transaction has been pruned, and in response, not executing the second transaction.

7. The medium of claim 1, wherein:

bytecode of the first program remains in the tamper-evident data store after pruning.

8. The medium of claim 1, the operations comprising:

before pruning and after receiving the request, determining that an amount of pruning within a threshold duration of time is less than a threshold.

9. The medium of claim 1, the operation comprising:

before pruning and after receiving the request, determining that an entity requesting pruning is authorized to cause the pruning based, in part, on which data is requested to be pruned.

10. The medium of claim 1, wherein pruning comprises:

changing local copies of the tamper-evident data store on each of a plurality of computing nodes of a decentralized computing platform with a local instance of a virtual machine executing at least part of the first program.

11. The medium of claim 10, wherein the at least part of the first program invokes an opcode of the virtual machine, the opcode being configured to receive as an argument an address of a smart contract to be pruned.

12. The medium of claim 11, wherein the opcode is configured to cause an event indicating pruning to be emitted during execution, the event including an identifier of the first program.
13. The medium of claim 1, wherein pruning comprises determining that pruned data is not referenced by another program among the plurality of programs.
14. The medium of claim 1, wherein pruning comprises recursing a graph of programs created by the first program or through programs created by the first program and pruning data of programs in the graph of programs from the tamper-evident data store.
15. The medium of claim 1, wherein:
the tamper-evident data store comprises a plurality of channels; and
the plurality of programs execute on decentralized computing platform that persists program state to the tamper-evident data store in association with corresponding ones of the channels and prevents cross-channel access to stored data.
16. The medium of claim 1, wherein:
tamper-evident data store stores transactions in a relational database and renders transactions tamper-evident through multi-party cryptographic signatures based on content of the transactions.
17. The medium of claim 1, wherein:
the first program is a smart contract of a Turing-complete decentralized computing platform;
the decentralized computing platform executes the smart contract on a plurality of peer computing nodes; and
the plurality of peer computing nodes use means for determining consensus to determine a result of the smart contract execution.

18. The medium of claim 1, the operations comprising:

executing a third program that transfers a cryptographic token from a first account to a second account or associates an unspent transaction output including the cryptographic token with an address of a user.

19. The medium of claim 1, wherein:

pruning comprises steps for pruning.

20. A method, comprising:

receiving, with a computer system, a request to prune data stored in a tamper-evident data store, wherein:

the data includes first program state of a first program at a first time,

the first program is one of a plurality of programs having program state stored in the tamper-evident data store,

the request uniquely identifies the first program with an identifier that distinguishes the first program from a second program among the plurality of programs,

the tamper-evident data store comprises a directed acyclic graph of cryptographic hash pointers,

a first subset of the cryptographic hash pointers have cryptographic hash values based on the first program state of the first program at the first time,

a second subset of the cryptographic hash pointers have cryptographic hash values based on second program state of the first program at a second time,

a third subset of the cryptographic hash pointers have cryptographic hash values based on the first program state of the first program at the first time, the second program state of the first program at the second time, and program state of the second program from before the first time, and

the tamper-evident data store comprises or is associated with a record that links the first subset of the cryptographic hash pointers to the second subset of the cryptographic hash pointers;

in response to the request, based on the identifier and the record that links the first subset of the cryptographic hash pointers to the second subset of the cryptographic hash pointers, with the computer system, selecting:

a first subset of nodes of the directed acyclic graph having first content based on the first program state of the first program at the first time, the hash values of first subset of the cryptographic hash pointers being based on the first content, and

a second subset of nodes of the directed acyclic graph having second content based on the first program state of the first program at the first time, the hash values of second subset of the cryptographic hash pointers being based on the second content;

pruning, with the computer system, the first program state and the second program state from the tamper-evident data store by changing the first content and the second content in memory, wherein:

after the pruning, cryptographic hash values of the directed acyclic graph of cryptographic hash pointers are internally consistent, and

after the pruning, program state of second program from before the first time remains in the tamper-evident data store.