

MULTIPLE-PHASE REWRITABLE BLOCKCHAIN

Priority

[001] This application claims priority to U.S. Provisional Patent Application Serial No. 62/553,597, filed on 1 September 2017, Attorney Docket No. 15718/278, titled Multiple-Phase Rewritable Blockchain, which is incorporated by reference in its entirety.

Technical Field

[002] This disclosure relates to data verification, validation, and re-writing in complex real-world systems.

Background

[003] Rapid advances in electronics and communication technologies, driven by immense customer demand, have resulted in the widespread adoption of electronic transactions and record keeping. As one example, e-currencies, such as Bitcoin, have displaced paper currencies in millions of transactions per year. Improvements in the verification and recordation of such electronic transactions will continue to increase the features and options available to operators engaging in electronic transactions.

Summary of Invention

[003a] According to a first aspect of the present invention, there is provided a system including: a memory configured to: store a blockchain secured by an integrity code; and blockchain phase circuitry configured to: during a rewrite-permissive phase for a trusted party: cause, on behalf of the trusted party, execution of a rewrite of a block within the blockchain, the rewrite coding-consistent with the integrity code; and responsive to a trigger condition, alter access to rewrites for the trusted party to initiate a rewrite-embargoed phase for the trusted party, the trigger condition including an update to software code for the blockchain; and during a rewrite-embargoed phase for the trusted

party: cause, on behalf of the trusted party, unsuccessful execution of a rewrite of a block within the blockchain, the rewrite coding-inconsistent with the integrity code.

[003b] According to a second aspect of the present invention, there is provided a method including: during a rewrite-permissive phase for a first trusted party, receiving a first command to perform a first validity-preserving rewrite to a blockchain; responsive to receiving the first command during the rewrite-permissive phase, causing execution of the first validity-preserving rewrite to the blockchain, the first validity-preserving rewrite coding-consistent with an existing integrity output stored within the blockchain; responsive to a trigger condition, altering access to validity-preserving rewrites to terminate the rewrite-permissive phase for the first trusted party, the trigger condition including an update to software code for the blockchain; during a rewrite-embargoed phase for a second trusted party, receiving a second command to perform a second validity-preserving rewrite to the blockchain; and responsive to receiving the second command during the rewrite-embargoed phase, causing unsuccessful execution of the second validity-preserving rewrite to the blockchain.

[003c] According to a third aspect of the present invention, there is provided a system including: memory configured to store a blockchain; and blockchain phase circuitry configured to: receive a first command to perform a first validity-preserving rewrite to a blockchain, the first command received during a rewrite-permissive phase for a first trusted party; responsive to receiving the first command during the rewrite-permissive phase, cause execution of the first validity-preserving rewrite to the blockchain, the first validity-preserving rewrite coding-consistent with an existing integrity output stored within the blockchain; responsive to a trigger condition, alter access to validity-preserving rewrites to terminate the rewrite-permissive phase for the first trusted party, the trigger condition including an update to software code for the blockchain; receive a second command to perform a second validity-preserving rewrite to the blockchain, the second command received during a rewrite-embargoed phase for a second trusted party; and responsive to receiving the second command during the rewrite-embargoed phase, fail to cause execution of the second validity-preserving rewrite to the blockchain.

BRIEF DESCRIPTION OF THE DRAWINGS

[004] Figure 1 shows two example views of a blockchain.

[005] Figure 2 shows two example rewrites to the example blockchain of Figure 1.

[006] Figure 3 shows an example blockchain processing system.

- [007]** Figure 4 shows an example blockchain rewriting system.
- [008]** Figure 5 shows example rewrite management logic.
- [009]** Figure 6 shows example rewrite logic.
- [010]** Figure 7A shows two example collision searches.
- [011]** Figure 7B shows an example rewrite to a blockchain using a collision.

Claims

What is claimed is:

1. A system including:
a memory configured to:
 - store a blockchain secured by an integrity code; andblockchain phase circuitry configured to:
 - during a rewrite-permissive phase for a trusted party:
 - cause, on behalf of the trusted party, execution of a rewrite of a block within the blockchain,
 - the rewrite coding-consistent with the integrity code; and
 - responsive to a trigger condition, alter access to rewrites for the trusted party to initiate a rewrite-embargoed phase for the trusted party, the trigger condition including an update to software code for the blockchain; and
 - during a rewrite-embargoed phase for the trusted party:
 - cause, on behalf of the trusted party, unsuccessful execution of a rewrite of a block within the blockchain,
 - the rewrite coding-inconsistent with the integrity code.
2. The system of claim 1, where the blockchain phase circuitry is configured to cause unsuccessful execution of the rewrite of the block by:
 - computing coding-inconsistent collision data based on less than complete access to a key secret for the integrity code.
3. The system of claim 1, where the blockchain phase circuitry is configured to cause unsuccessful execution of the rewrite of the block by:
 - denying rewrite privileges to the trusted party.
4. The system of claim 1, where the blockchain phase circuitry is configured to initiate the rewrite embargoed phase by deleting a key secret for the integrity code to.
5. The system of claim 1, where initiation of the rewrite-embargoed phase includes a permanent transformation of the blockchain to an immutable state.

6. A method including:

 during a rewrite-permissive phase for a first trusted party, receiving a first command to perform a first validity-preserving rewrite to a blockchain;

 responsive to receiving the first command during the rewrite-permissive phase, causing execution of the first validity-preserving rewrite to the blockchain, the first validity-preserving rewrite coding-consistent with an existing integrity output stored within the blockchain;

 responsive to a trigger condition, altering access to validity-preserving rewrites to terminate the rewrite-permissive phase for the first trusted party,

 the trigger condition including an update to software code for the blockchain;

 during a rewrite-embargoed phase for a second trusted party, receiving a second command to perform a second validity-preserving rewrite to the blockchain; and

 responsive to receiving the second command during the rewrite-embargoed phase, causing unsuccessful execution of the second validity-preserving rewrite to the blockchain.

7. The method of claim 6, where causing unsuccessful execution of the second validity-preserving rewrite to the blockchain includes:

 computing coding-inconsistent collision data based on incomplete access to a key secret for an integrity code of the blockchain.

8. The method of claim 6, where causing unsuccessful execution of the second validity-preserving rewrite to the blockchain includes:

 denying rewrite privileges to the second trusted party.

9. The method of claim 6, where altering access to validity-preserving rewrites includes deleting a key secret for an integrity code of the blockchain.

10. The method of claim 6, where the first trusted party and the second trusted party include the same trusted party.

11. The method of claim 6, where the first trusted party and the second trusted party include different trusted parties.

12. The method of claim 6, where the rewrite-permissive phase for the first trust party and the rewrite-embargoed phase for the second trusted party coincide at least in part.

13. The method of claim 6, where the rewrite-embargoed phase includes:
a permanent transformation of the blockchain to an immutable state.

14. A system including:

- memory configured to store a blockchain; and
- blockchain phase circuitry configured to:

- receive a first command to perform a first validity-preserving rewrite to a blockchain,

- the first command received during a rewrite-permissive phase for a first trusted party;

- responsive to receiving the first command during the rewrite-permissive phase, cause execution of the first validity-preserving rewrite to the blockchain,

- the first validity-preserving rewrite coding-consistent with an existing integrity output stored within the blockchain;

- responsive to a trigger condition, alter access to validity-preserving rewrites to terminate the rewrite-permissive phase for the first trusted party,

- the trigger condition including an update to software code for the blockchain;

- receive a second command to perform a second validity-preserving rewrite to the blockchain,

- the second command received during a rewrite-embargoed phase for a second trusted party; and

- responsive to receiving the second command during the rewrite-embargoed phase, fail to cause execution of the second validity-preserving rewrite to the blockchain.

15. The system of claim 14, where the blockchain phase circuitry is configured to fail to cause execution of the second validity-preserving rewrite to the blockchain by:

- computing coding-inconsistent collision data based on less than complete access to a key secret for an integrity code of the blockchain.

z

16. The system of claim 14, where the blockchain phase circuitry is configured to fail to cause execution of the second validity-preserving rewrite to the blockchain by:

denying rewrite privileges to the second trusted party.

17. The system of claim 14, where the blockchain phase circuitry is configured to alter access to validity-preserving rewrites includes deleting a key secret for an integrity code of the blockchain.

18. The system of claim 17, where deletion a key secret includes deleting a decryption key used to decipher the key secret from an encrypted form.

19. The system of claim 14, where the rewrite-permissive phase for the first trusted party and the rewrite-embargoed phase for the second trusted party coincide at least in part.

20. The system of claim 14, where the rewrite-embargoed phase includes:
a permanent transformation of the blockchain to an immutable state.