

TURN-CONTROL 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 invention, there is provided a system including: memory configured to store a blockchain; and blockchain phase circuitry configured to: initiate a rewrite-permissive phase for a selected trusted party by obtaining, on behalf of the selected trusted party, access that allows combination of a turn-control portion of a key secret with a first portion of the key secret, the first portion controlled by the selected trusted party; combine the turn-control portion and the first portion to generate the key secret; receive a first command to perform a first validity-preserving rewrite to the blockchain, the first command received during the rewrite-permissive phase for the selected trusted party, and the first validity-preserving rewrite coding-consistent with an

existing integrity output stored within the blockchain; responsive to receiving the first command during the rewrite-permissive phase, cause execution of the first validity-preserving rewrite to the blockchain using the key secret; after the rewrite-permissive phase for the selected trusted party, initiate a rewrite embargoed phase by altering the access in response to request for write privileges from a specific trusted party different from the selected trusted party; after altering the access, fail to combine the first portion and the turn-control portion during a rewrite-embargoed phase for the selected trusted party; receive a second command to perform a second validity-preserving rewrite to the blockchain, the second command received during the rewrite-embargoed phase for the selected 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.

[003b] According to a second aspect of the present invention, there is provided a method including: initiating a rewrite-permissive phase for a selected trusted party by obtaining, on behalf of the selected trusted party, access that allows combination of a turn-control portion of a key secret with a first portion of the key secret, the first portion controlled by the selected trusted party; combining the turn-control portion and the first portion to generate the key secret; receiving a first command on behalf of the selected trusted party, the first command to perform a first validity-preserving rewrite to a blockchain, the first command received during the rewrite-permissive phase for the selected trusted party; responsive to receiving the first command during the rewrite-permissive phase, causing execution of the first validity-preserving rewrite to the blockchain using the key secret, the first validity-preserving rewrite coding-consistent with a first existing integrity output stored within the blockchain; after the rewrite-permissive phase for the selected trusted party: initiating a rewrite-embargoed phase for the selected trusted party by altering the access to allow combination of the turn-control portion with a second portion of the key secret, the second portion of the key secret controlled by a target trusted party different than the selected trusted party; after altering the access, failing to combine the first portion and the turn-control portion during the rewrite-embargoed phase for the selected trusted party; receiving a second command on behalf of the selected trusted party, the second command to perform a second validity-preserving rewrite to the blockchain, the second command received during the rewrite-embargoed phase for the selected trusted party; responsive to receiving the second command during the rewrite-embargoed phase, failing to cause execution of the

second validity-preserving rewrite to the blockchain; receiving a third command on behalf of the target trusted party, the third command to perform a third validity-preserving rewrite to the blockchain, the third command received during the rewrite-embargoed phase for the selected trusted party; and responsive to receiving the third command during the rewrite-embargoed phase, causing execution of the second validity-preserving rewrite to the blockchain using the key secret, the second validity-preserving rewrite coding-consistent with a second existing integrity output stored within 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: initiate a rewrite-permissive phase for a selected trusted party by obtaining, on behalf of the selected trusted party, access that allows combination of a turn-control portion of a key secret with a first portion of the key secret, the first portion controlled by the selected trusted party; combine the turn-control portion and the first portion to generate the key secret; receive a first command on behalf of the selected trusted party, the first command to perform a first validity-preserving rewrite to the blockchain, the first command received during the rewrite-permissive phase for the selected trusted party; responsive to receiving the first command during the rewrite-permissive phase, cause execution of the first validity-preserving rewrite to the blockchain using the key secret, the first validity-preserving rewrite coding-consistent with a first existing integrity output stored within the blockchain; after the rewrite-permissive phase for the selected trusted party: initiate a rewrite-embargoed phase for the selected trusted party by altering the access to allow combination of the turn-control portion with a second portion of the key secret, the second portion of the key secret controlled by a target trusted party different than the selected trusted party; after altering the access, fail to combine the first portion and the turn-control portion during the rewrite-embargoed phase for the selected trusted party; receive a second command on behalf of the selected trusted party, the second command to perform a second validity-preserving rewrite to the blockchain, the second command received during the rewrite-embargoed phase for the selected trusted party; responsive to receiving the second command during the rewrite-embargoed phase, fail to cause execution of the second validity-preserving rewrite to the blockchain; receive a third command on behalf of the target trusted party, the third command to perform a third validity-preserving rewrite to the blockchain, the third command received during the

rewrite-embargoed phase for the selected trusted party; and responsive to receiving the third command during the rewrite-embargoed phase, cause execution of the second validity-preserving rewrite to the blockchain using the key secret, the second validity-preserving rewrite coding-consistent with a second existing integrity output stored within 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:

memory configured to store a blockchain; and

blockchain phase circuitry configured to:

initiate a rewrite-permissive phase for a selected trusted party by obtaining, on behalf of the selected trusted party, access that allows combination of a turn-control portion of a key secret with a first portion of the key secret,

the first portion controlled by the selected trusted party;

combine the turn-control portion and the first portion to generate the key secret;

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

the first command received during the rewrite-permissive phase for the selected trusted party, and

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

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

after the rewrite-permissive phase for the selected trusted party, initiate a rewrite embargoed phase by altering the access in response to request for write privileges from a specific trusted party different from the selected trusted party;

after altering the access, fail to combine the first portion and the turn-control portion during a rewrite-embargoed phase for the selected trusted party;

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

the second command received during the rewrite-embargoed phase for the selected 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.

2. The system of claim 1, 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 after altering the access and failure to combine the first portion and the turn-control portion during the rewrite-embargoed phase.

3. The system of claim 1, where the blockchain phase circuitry is configured to cause unsuccessful execution of the second validity-preserving rewrite of the blockchain by:

denying rewrite privileges to the selected trusted party during the rewrite-embargoed phase.

4. The system of claim 1, where the blockchain phase circuitry is configured to initiate the rewrite-embargoed phase for the selected trusted party by altering the access to allow combination of the turn-control portion with a second portion of the key secret,

the second portion of the key secret controlled by a target trusted party different than the selected trusted party.

5. The system of claim 4, where, after the rewrite-embargoed phase for the selected trusted party, the blockchain phase circuitry is configured to initiate a rewrite-embargoed phase for the target trusted party by altering the access to allow combination of the turn-control portion with a third portion of the key secret,

the third portion of the key secret controlled by a designated trusted party different than the selected trusted party and the target trusted party.

6. The system of claim 1, where the blockchain phase circuitry is configured to allow concurrent rewrite-permissive phases for the selected trusted party and a target trusted party by providing the access to allow combination of the turn-control portion with a second portion of the key secret during the rewrite-permissive phase for the selected trusted party,

the second portion of the key secret controlled by a target trusted party, and

the target trusted party different than the selected trusted party.

7. The system of claim 1, where the blockchain phase circuitry is configured to, after the rewrite-embargoed phase, re-obtain, on behalf of the selected trusted party, the access

that allows combination of a turn-control portion of a key secret with a first portion of the key secret to initiate an additional rewrite-permissive phase of the selected trusted party.

8. The system of claim 1, where the blockchain phase circuitry is configured to pass the access among multiple trusted parties to execute a turn sequence,

the turn sequence defining rewrite-permissive phases and rewrite-embargoed phases for individual ones of the multiple trusted parties, and

the multiple trusted parties including the selected trusted party.

9. The system of claim 8, where the blockchain phase circuitry is configured to delete the turn-control portion to terminate the turn sequence.

10. The system of claim 8, where the blockchain phase circuitry is configured to temporarily deny the access to all of the multiple trusted parties to execute a break in the turn sequence.

11. The system of claim 1, where the blockchain phase circuitry is configured to combine the turn-control portion and the first portion via a key exchange operation.

12. The system of claim 11, where the key exchange operation is configured to combine turn-control portion and the first portion without disclosing content of the turn-control portion to the selected trusted party.

13. The system of claim 1, where the blockchain phase circuitry is configured to initiate the rewrite-embargoed phase for the selected trusted party responsive to a trigger condition.

14. The system of claim 13, where the trigger condition includes an expiry of a timer, a received command indicating a phase change, a detected software update, a request for a turn end, or any combination thereof.

15. A method including:

- initiating a rewrite-permissive phase for a selected trusted party by obtaining, on behalf of the selected trusted party, access that allows combination of a turn-control portion of a key secret with a first portion of the key secret,

 - the first portion controlled by the selected trusted party;

 - combining the turn-control portion and the first portion to generate the key secret;

- receiving a first command on behalf of the selected trusted party, the first command to perform a first validity-preserving rewrite to a blockchain,

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

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

 - after the rewrite-permissive phase for the selected trusted party:

 - initiating a rewrite-embargoed phase for the selected trusted party by altering the access to allow combination of the turn-control portion with a second portion of the key secret,

 - the second portion of the key secret controlled by a target trusted party different than the selected trusted party;

 - after altering the access, failing to combine the first portion and the turn-control portion during the rewrite-embargoed phase for the selected trusted party;

- receiving a second command on behalf of the selected trusted party, the second command to perform a second validity-preserving rewrite to the blockchain,

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

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

- receiving a third command on behalf of the target trusted party, the third command to perform a third validity-preserving rewrite to the blockchain,

 - the third command received during the rewrite-embargoed phase for the selected trusted party; and

responsive to receiving the third command during the rewrite-embargoed phase, causing execution of the second validity-preserving rewrite to the blockchain using the key secret, the second validity-preserving rewrite coding-consistent with a second existing integrity output stored within the blockchain.

16. The method of claim 15, further including passing the access among multiple trusted parties to execute a turn sequence,

the turn sequence defining rewrite-permissive phases and rewrite-embargoed phases for individual ones of the multiple trusted parties, and
the multiple trusted parties including the selected trusted party.

17. The method of claim 16, further including deleting the turn-control portion to terminate the turn sequence.

18. The method of claim 16, further including temporarily denying the access to all of the multiple trusted parties to execute a break in the turn sequence.

19. A system including:

memory configured to store a blockchain; and

blockchain phase circuitry configured to:

initiate a rewrite-permissive phase for a selected trusted party by obtaining, on behalf of the selected trusted party, access that allows combination of a turn-control portion of a key secret with a first portion of the key secret,

the first portion controlled by the selected trusted party;

combine the turn-control portion and the first portion to generate the key secret;

receive a first command on behalf of the selected trusted party, the first command to perform a first validity-preserving rewrite to the blockchain,

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

responsive to receiving the first command during the rewrite-permissive phase, cause execution of the first validity-preserving rewrite to the blockchain using the key secret, the first validity-preserving rewrite coding-consistent with a first existing integrity output stored within the blockchain;

after the rewrite-permissive phase for the selected trusted party:

initiate a rewrite-embargoed phase for the selected trusted party by altering the access to allow combination of the turn-control portion with a second portion of the key secret,

the second portion of the key secret controlled by a target trusted party different than the selected trusted party;

after altering the access, fail to combine the first portion and the turn-control portion during the rewrite-embargoed phase for the selected trusted party;

receive a second command on behalf of the selected trusted party, the second command to perform a second validity-preserving rewrite to the blockchain, the second command received during the rewrite-embargoed phase for the selected trusted party;

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

receive a third command on behalf of the target trusted party, the third command to perform a third validity-preserving rewrite to the blockchain,

the third command received during the rewrite-embargoed phase for the selected trusted party; and

responsive to receiving the third command during the rewrite-embargoed phase, cause execution of the second validity-preserving rewrite to the blockchain using the key secret, the second validity-preserving rewrite coding-consistent with a second existing integrity output stored within the blockchain.

20. The system of claim 19, where the blockchain phase circuitry is configured to pass the access among multiple trusted parties to execute a turn sequence,

the turn sequence defining rewrite-permissive phases and rewrite-embargoed phases for individual ones of the multiple trusted parties, and

the multiple trusted parties including the selected trusted party.