

CRYPTOGRAPHIC DATASHARE CONTROL FOR BLOCKCHAIN

Priority

5 [001] This application also claims priority to U.S. Patent Application No. 16/672,761, filed 4 November 2019, Attorney Docket No. 15718-511, which is incorporated herein in its entirety. This application also claims priority to U.S. Provisional Patent Application No. 62/757,273, filed 8 November 2018, Attorney Docket No. 15718-499, which is incorporated herein in its entirety.

10

Technical Field

[002] This disclosure relates to controlled datasharing in distributed network environments.

15 Background

[003] Rapid advances in electronics and communication technologies, driven by immense customer demand, have resulted in the widespread adoption of electronic transactions, record keeping, and data sharing. As one example, e-currencies, such as Bitcoin, have displaced paper currencies in millions of transactions per year.

20 Improvements in tracking and/or knowledge management attached to such electronic transactions will continue to increase the features and options available to operators engaging in electronic transactions.

Summary

25 [003a] According to a first aspect of the present invention, there is provided a method including: selecting underlying compute data for provision to a distributed ledger technology (DLT) based data construct; encrypting the underlying compute data to generate homomorphic compute data with a homomorphic layer of encryption, the underlying compute data encrypted in accord with a homomorphic encryption scheme;

30 determining a compute party associated with the DLT-based data construct, the compute party selected to cause performance of one or more processing operations on the homomorphic compute data, and the one or more processing operations performed without concurrent decrypted access to the underlying compute data; encrypting the homomorphic compute data with a public key to generate secured homomorphic

compute data by adding a secured layer of public key encryption above the homomorphic layer, the public key associated with an originating party; causing the secured homomorphic compute data to be added to a current block of the DLT-based data construct, the adding to the current block of the DLT-based data construct to effect
5 a controlled share of the homomorphic compute data with the compute party, and addition of the secured homomorphic compute data to the current block including: generation of a previous hash of a previous block; placement of the previous hash and the secured homomorphic compute data in the current block; and addition of the current block to the DLT-based data construct; and authorizing access, by the compute party, to
10 the homomorphic compute data to perform the one or more processing operations while the homomorphic compute data is still encrypted under the homomorphic layer of encryption, the access authorized by decrypting the secured homomorphic compute data to remove the secured layer.

[003b] According to a second aspect of the present invention, there is provided a
15 system including: network interface circuitry configured to establish a communication link with a distributed network supporting a distributed ledger technology (DLT) based data construct; and controlled datashare circuitry configured to: select underlying compute data for provision to a distributed ledger technology (DLT) based data construct; encrypt the underlying compute data in accord with a homomorphic
20 encryption scheme to generate homomorphic compute data with a homomorphic layer of encryption; determine a compute party associated with the DLT-based data construct, the compute party selected to cause performance of one or more processing operations on the homomorphic compute data without concurrent decrypted access to the underlying compute data; encrypt the homomorphic compute data with a public key
25 associated with an originating party to generate secured homomorphic compute data by adding a secured layer of public key encryption above the homomorphic layer; cause the secured homomorphic compute data to be added to a current block of the DLT-based data construct to effect a controlled share of the homomorphic compute data with the compute party, addition of the secured homomorphic compute data to the current
30 block including: generation of a previous hash of a previous block; placement of the previous hash and the secured homomorphic compute data in the current block; and addition of the current block to the DLT-based data construct; and authorize access, by the compute party, to the homomorphic compute data to perform the one or more processing operations while the homomorphic compute data is still encrypted under the

homomorphic layer of encryption by decrypting the secured homomorphic compute data to remove the secured layer.

[003c] According to a third aspect of the present invention, there is provided a product including: machine-readable media other than a transitory signal; instructions stored on the machine-readable media, the instructions configured to, when executed, cause a machine to: select underlying compute data for provision to a distributed ledger technology (DLT) based data construct; encrypt the underlying compute data in accord with a homomorphic encryption scheme to generate homomorphic compute data with a homomorphic layer of encryption; determine a compute party associated with the DLT-based data construct, the compute party selected to cause performance of one or more processing operations on the homomorphic compute data without concurrent decrypted access to the underlying compute data; encrypt the homomorphic compute data with a public key associated with an originating party to generate secured homomorphic compute data by adding a secured layer of public key encryption above the homomorphic layer; cause the secured homomorphic compute data to be added to a current block of the DLT-based data construct to effect a controlled share of the homomorphic compute data with the compute party, addition of the secured homomorphic compute data to the current block including: generation of a previous hash of a previous block; placement of the previous hash and the secured homomorphic compute data in the current block; and addition of the current block to the DLT-based data construct; and authorize access, by the compute party, to the homomorphic compute data to perform the one or more processing operations while the homomorphic compute data is still encrypted under the homomorphic layer of encryption by decrypting the secured homomorphic compute data to remove the secured layer.

BRIEF DESCRIPTION OF THE DRAWINGS

[004] Figure 1 shows an example controlled datashare environment.

[005] Figure 2 shows example controlled datashare logic

[006] Figure 3 shows an example datashare control system.

[007] Figure 4 shows an example system for writing data to a blockchain.

[008] Figure 5 shows an example system to support application of processing operations to homomorphically encrypted data.

[009] Figure 6 shows an example system to support data sharing.

Claims

What is claimed is:

1. A method including:
 - 5 selecting underlying compute data for provision to a distributed ledger technology (DLT) based data construct;
encrypting the underlying compute data to generate homomorphic compute data with a homomorphic layer of encryption,
the underlying compute data encrypted in accord with a homomorphic encryption
10 scheme;
determining a compute party associated with the DLT-based data construct,
the compute party selected to cause performance of one or more processing operations on the homomorphic compute data, and
the one or more processing operations performed without concurrent decrypted
15 access to the underlying compute data;
encrypting the homomorphic compute data with a public key to generate secured homomorphic compute data by adding a secured layer of public key encryption above the homomorphic layer,
the public key associated with an originating party;
20 causing the secured homomorphic compute data to be added to a current block of the DLT-based data construct,
the adding to the current block of the DLT-based data construct to effect a controlled share of the homomorphic compute data with the compute party, and
addition of the secured homomorphic compute data to the current block
25 including:
generation of a previous hash of a previous block;
placement of the previous hash and the secured homomorphic compute data in the current block; and
addition of the current block to the DLT-based data construct; and
30 authorizing access, by the compute party, to the homomorphic compute data to perform the one or more processing operations while the homomorphic compute data is still encrypted under the homomorphic layer of encryption,
the access authorized by decrypting the secured homomorphic compute data to remove the secured layer.

2. The method of claim 1, further including gleaning derived insights by decrypting the homomorphic compute data after machine learning is applied via the one or more processing operations.

5

3. The method of claim 1, where the compute party is one of multiple parties that direct performance of processing operations on encrypted forms of the underlying compute data.

10 4. The method of claim 1, where decrypting the secured homomorphic compute data includes using a private key controlled by the originator party,
the private key including an asymmetrical pair to the public key.

15 5. The method of claim 4, where the compute party causes performance of the processing operations on the homomorphic compute data while encrypted using the homomorphic encryption scheme.

6. The method of claim 1, where the one or more processing operations include a multiplication operation (MULT), an addition operation (ADD), or a combination thereof.

20

7. The method of claim 1, where effecting the controlled share of the underlying compute data does not disclose underlying values within the underlying compute data to the compute party.

25 8. The method of claim 1, further including:
receiving homomorphic processed data, the homomorphic processed data generated by performing the one or more processing operations on the homomorphic computed data on behalf the compute party.

30 9. The method of claim 8, further including decrypting the homomorphic processed data using identical decryption parameters to those which would successfully decrypt the homomorphic compute data to access underlying processed data.

10. The method of claim 9, further including sharing the underlying processed data by coordinating authorization with the compute party.

11. The method of claim 10, further including effecting the sharing of the underlying processed data by appending the underlying processed data to the DLT-based data construct in a sharing block subsequent to the current block.

12. The method of claim 1, further including confirming that the underlying compute data is the same across multiple instances of the DLT-based data construct.

13. A system including:

network interface circuitry configured to establish a communication link with a distributed network supporting a distributed ledger technology (DLT) based data construct; and

controlled datashare circuitry configured to:

select underlying compute data for provision to a distributed ledger technology (DLT) based data construct;

encrypt the underlying compute data in accord with a homomorphic encryption scheme to generate homomorphic compute data with a homomorphic layer of encryption;

determine a compute party associated with the DLT-based data construct, the compute party selected to cause performance of one or more processing operations on the homomorphic compute data without concurrent decrypted access to the underlying compute data;

encrypt the homomorphic compute data with a public key associated with an originating party to generate secured homomorphic compute data by adding a secured layer of public key encryption above the homomorphic layer;

cause the secured homomorphic compute data to be added to a current block of the DLT-based data construct to effect a controlled share of the homomorphic compute data with the compute party, addition of the secured homomorphic compute data to the current block including:

generation of a previous hash of a previous block;

placement of the previous hash and the secured homomorphic compute data in the current block; and

addition of the current block to the DLT-based data construct; and
authorize access, by the compute party, to the homomorphic compute
data to perform the one or more processing operations while the homomorphic
compute data is still encrypted under the homomorphic layer of encryption by
5 decrypting the secured homomorphic compute data to remove the secured layer.

14. The system claim 13, where the controlled datashare circuitry is further
configured to glean derived insights by decrypting the homomorphic compute data after
machine learning is applied via the one or more processing operations.

10 15. The system of claim 13, where the compute party is one of multiple parties that
direct performance of processing operations on encrypted forms of the underlying
compute data.

16. The system of claim 13, where the controlled datashare circuitry is configured to
decrypt the secured homomorphic compute data includes using a private key controlled
by the originator party,
the private key including an asymmetrical pair to the public key.

20 17. The system of claim 16, where the compute party causes performance of the
processing operations on the homomorphic compute data while encrypted using the
homomorphic encryption scheme.

18. The system of claim 13, where the one or more processing operations include a
25 multiplication operation (MULT), an addition operation (ADD), or a combination thereof.

19. The system of claim 13, where the controlled datashare circuitry is configured to
effect the controlled share of the underlying compute data without causing disclosure of
underlying values within the underlying compute data to the compute party.

30 20. A product including:
machine-readable media other than a transitory signal;
instructions stored on the machine-readable media, the instructions configured
to, when executed, cause a machine to:

select underlying compute data for provision to a distributed ledger technology (DLT) based data construct;

encrypt the underlying compute data in accord with a homomorphic encryption scheme to generate homomorphic compute data with a homomorphic layer of encryption;

determine a compute party associated with the DLT-based data construct, the compute party selected to cause performance of one or more processing operations on the homomorphic compute data without concurrent decrypted access to the underlying compute data;

encrypt the homomorphic compute data with a public key associated with an originating party to generate secured homomorphic compute data by adding a secured layer of public key encryption above the homomorphic layer;

cause the secured homomorphic compute data to be added to a current block of the DLT-based data construct to effect a controlled share of the homomorphic compute data with the compute party, addition of the secured homomorphic compute data to the current block including:

generation of a previous hash of a previous block;

placement of the previous hash and the secured homomorphic compute data in the current block; and

addition of the current block to the DLT-based data construct; and

authorize access, by the compute party, to the homomorphic compute data to perform the one or more processing operations while the homomorphic compute data is still encrypted under the homomorphic layer of encryption by decrypting the secured homomorphic compute data to remove the secured layer.