

CLAIMS

1. A method for token generation using SMPC compute engines, the method comprising:
 - hashing, by a node, a data input with a salt value;
 - splitting, by the node, the hashed data input into a plurality of secret shares, wherein each respective secret share of the plurality of secret shares is assigned to a respective SMPC compute engine of a plurality of SMPC compute engines,
 - wherein the respective SMPC compute engine is configured to:
 - securely hash the respective secret share with a secret salt value unique to the respective SMPC compute engine;
 - transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;
 - receive a plurality of hashed secret shares from remaining SMPC compute engines of the plurality of SMPC compute engines; and
 - generate a token, wherein the token is a combination of the hashed respective secret share and the plurality of hashed secret shares.
2. The method of claim 1, wherein the node is one data source of multiple data sources, wherein another token corresponding to data inputs from both the node and at least one other node of the multiple data sources should be generated, further comprising:
 - transmitting the salt value to the at least one other node, wherein the at least one other node is configured to:
 - hash at least one other data input with the salt value; and
 - split the hashed at least one other data input into at least one other plurality of secret shares, wherein each respective secret share of the at least one other plurality of secret shares is assigned to a respective SMPC compute engine of the plurality of SMPC compute engines.
3. The method of claim 2, wherein the respective SMPC compute engine is further configured to:

jointly, with the plurality of SMPC compute engines, hash the respective secret share from the plurality of secret shares with a secret salt value β and hash the respective secret share from the at least one other plurality of secret shares with the secret salt value β , where β is unknown to any SMPC compute engine of the plurality of SMPC compute engines;

transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive another plurality of hashed secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate additional tokens, wherein each additional token is a combination of the plurality of hashed secret shares, irrespective of data source.

4. The method of claim 3, wherein the node and the at least one other node all receive the additional tokens.

5. The method of claim 1, wherein the data input is an identifier of a respective row or column in a dataset being uploaded by the node to the plurality of SMPC compute engines.

6. The method of claim 1, further comprising:

prior to splitting the hashed data input, combining the hashed data input with a passcode provided by a user of the node; and

splitting the hashed data input combined with the passcode into the plurality of secret shares.

7. The method of claim 1, wherein the respective SMPC compute engine is further configured to:

securely encrypt the respective secret share using an encryption scheme, wherein an initialization vector and key of the encryption scheme are in a secret share and are unknown to any SMPC compute engine of the plurality of SMPC compute engines;

transmit the respective encrypted secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive a plurality of encrypted secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate the token, wherein the token is a combination of the encrypted respective secret share and the plurality of encrypted secret shares.

8. The method of claim 1, wherein the respective SMPC compute engine is further configured to:

store a plurality of generated tokens with corresponding data inputs;

in response to detecting that the stored plurality of generated tokens is being overwritten, generate an alert indicating malicious behaviour.

9. A system for token generation using SMPC compute engines, the system comprising:

a first hardware processor of a node;

a second hardware processor of an SMPC compute engine;

the first hardware processor configured to:

hash a data input with a salt value;

split the hashed data input into a plurality of secret shares, wherein each respective secret share of the plurality of secret shares is assigned to a respective SMPC compute engine of a plurality of SMPC compute engines,

wherein the second hardware processor of the respective SMPC compute engine is configured to:

securely hash the respective secret share with a secret salt value unique to the respective SMPC compute engine;

transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive a plurality of hashed secret shares from remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate a token, wherein the token is a combination of the hashed respective secret share and the plurality of hashed secret shares.

10. The system of claim 9, wherein the node is one data source of multiple data sources, wherein another token corresponding to data inputs from both the node and at least one other node of the multiple data sources should be generated, wherein the first hardware processor is further configured to:

transmit the salt value to the at least one other node, wherein a third hardware processor of the at least one other node is configured to:

hash at least one other data input with the salt value; and

split the hashed at least one other data input into at least one other plurality of secret shares, wherein each respective secret share of the at least one other plurality of secret shares is assigned to a respective SMPC compute engine of the plurality of SMPC compute engines.

11. The system of claim 10, wherein the second hardware processor of the respective SMPC compute engine is further configured to:

jointly, with the plurality of SMPC compute engines, hash the respective secret share from the plurality of secret shares with a secret salt value β and hash the respective secret share from the at least one other plurality of secret shares with the secret salt value β , where β is unknown to any SMPC compute engine of the plurality of SMPC compute engines;

transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive another plurality of hashed secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate additional tokens, wherein each additional token is a combination of the plurality of hashed secret shares, irrespective of data source.

12. The system of claim 11, wherein the node and the at least one other node all receive the additional tokens.

13. The system of claim 9, wherein the data input is an identifier of a respective row or column in a dataset being uploaded by the node to the plurality of SMPC compute engines.

14. The system of claim 9, wherein the first hardware processor is further configured to:
prior to splitting the hashed data input, combine the hashed data input with a passcode provided by a user of the node; and
split the hashed data input combined with the passcode into the plurality of secret shares.
15. The system of claim 9, wherein the second hardware processor of the respective SMPC compute engine is further configured to:
securely encrypt the respective secret share using an encryption scheme, wherein an initialization vector and key of the encryption scheme are in a secret share and are unknown to any SMPC compute engine of the plurality of SMPC compute engines;
transmit the respective encrypted secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;
receive a plurality of encrypted secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and
generate the token, wherein the token is a combination of the encrypted respective secret share and the plurality of encrypted secret shares.
16. The system of claim 9, wherein the respective SMPC compute engine is further configured to:
store a plurality of generated tokens with corresponding data inputs;
in response to detecting that the stored plurality of generated tokens is being overwritten, generate an alert indicating malicious behaviour.
17. A non-transitory computer readable medium storing thereon computer executable instructions for token generation using SMPC compute engines, comprising instructions for:
hashing, by a node, a data input with a salt value;
splitting, by the node, the hashed data input into a plurality of secret shares, wherein each respective secret share of the plurality of secret shares is assigned to a respective SMPC compute engine of a plurality of SMPC compute engines,
wherein the respective SMPC compute engine is configured to:

securely hash the respective secret share with a secret salt value unique to the respective SMPC compute engine;

transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive a plurality of hashed secret shares from remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate a token, wherein the token is a combination of the hashed respective secret share and the plurality of hashed secret shares.

18. The non-transitory computer readable medium of claim 17, wherein the node is one data source of multiple data sources, wherein another token corresponding to data inputs from both the node and at least one other node of the multiple data sources should be generated, further comprising instructions for:

transmitting the salt value to the at least one other node, wherein the at least one other node is configured to:

hash at least one other data input with the salt value; and

split the hashed at least one other data input into at least one other plurality of secret shares, wherein each respective secret share of the at least one other plurality of secret shares is assigned to a respective SMPC compute engine of the plurality of SMPC compute engines.

19. The non-transitory computer readable medium of claim 15, wherein the data input is an identifier of a respective row or column in a dataset being uploaded by the node to the plurality of SMPC compute engines.

20. The non-transitory computer readable medium of claim 17, further comprising instructions for:

prior to splitting the hashed data input, combining the hashed data input with a passcode provided by a user of the node; and

splitting the hashed data input combined with the passcode into the plurality of secret shares.

CLAIMS

1. A method for token generation using secure multiparty computation (SMPC) compute engines, the method comprising:
 - hashing, by a node, a data input with a salt value;
 - combining the hashed data input with a passcode provided by a user of the node;
 - splitting, by the node, the hashed data input combined with the passcode into a plurality of secret shares, wherein each respective secret share of the plurality of secret shares is assigned to a respective SMPC compute engine of a plurality of SMPC compute engines,
 - wherein the respective SMPC compute engine is configured to:
 - securely hash the respective secret share with a secret salt value unique to the respective SMPC compute engine;
 - transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;
 - receive a plurality of hashed secret shares from remaining SMPC compute engines of the plurality of SMPC compute engines;
 - generate a token, wherein the token is a combination of the hashed respective secret share and the plurality of hashed secret shares;
 - store a plurality of generated tokens with corresponding data inputs; and
 - in response to detecting that the stored plurality of generated tokens is being overwritten, generate an alert indicating malicious behaviour.
2. The method of claim 1, wherein the node is one data source of multiple data sources, wherein another token corresponding to data inputs from both the node and at least one other node of the multiple data sources should be generated, further comprising:
 - transmitting the salt value to the at least one other node, wherein the at least one other node is configured to:
 - hash at least one other data input with the salt value; and
 - split the hashed at least one other data input into at least one other plurality of secret shares, wherein each respective secret share of the at least one other plurality of

secret shares is assigned to a respective SMPC compute engine of the plurality of SMPC compute engines.

3. The method of claim 2, wherein the respective SMPC compute engine is further configured to:

jointly, with the plurality of SMPC compute engines, hash the respective secret share from the plurality of secret shares with a secret salt value β and hash the respective secret share from the at least one other plurality of secret shares with the secret salt value β , where β is unknown to any SMPC compute engine of the plurality of SMPC compute engines;

transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive another plurality of hashed secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate additional tokens, wherein each additional token is a combination of the plurality of hashed secret shares, irrespective of data source.

4. The method of claim 3, wherein the node and the at least one other node all receive the additional tokens.

5. The method of claim 1, wherein the data input is an identifier of a respective row or column in a dataset being uploaded by the node to the plurality of SMPC compute engines.

6. The method of claim 1, wherein the respective SMPC compute engine is further configured to:

securely encrypt the respective secret share using an encryption scheme, wherein an initialization vector and key of the encryption scheme are in a secret share and are unknown to any SMPC compute engine of the plurality of SMPC compute engines;

transmit the respective encrypted secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive a plurality of encrypted secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate the token, wherein the token is a combination of the encrypted respective secret share and the plurality of encrypted secret shares.

7. A system for token generation using secure multiparty computation (SMPC) compute engines, the system comprising:

a first hardware processor of a node;

a second hardware processor of an SMPC compute engine;

the first hardware processor configured to:

hash a data input with a salt value;

combine the hashed data input with a passcode provided by a user of the node;

split the hashed data input combined with the passcode into a plurality of secret shares, wherein each respective secret share of the plurality of secret shares is assigned to a respective SMPC compute engine of a plurality of SMPC compute engines,

wherein the second hardware processor of the respective SMPC compute engine is configured to:

securely hash the respective secret share with a secret salt value unique to the respective SMPC compute engine;

transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive a plurality of hashed secret shares from remaining SMPC compute engines of the plurality of SMPC compute engines;

generate a token, wherein the token is a combination of the hashed respective secret share and the plurality of hashed secret shares;

store a plurality of generated tokens with corresponding data inputs; and

in response to detecting that the stored plurality of generated tokens is being overwritten, generate an alert indicating malicious behaviour.

8. The system of claim 7, wherein the node is one data source of multiple data sources, wherein another token corresponding to data inputs from both the node and at least one other node of the multiple data sources should be generated, wherein the first hardware processor is further configured to:

transmit the salt value to the at least one other node, wherein a third hardware processor of the at least one other node is configured to:

hash at least one other data input with the salt value; and

split the hashed at least one other data input into at least one other plurality of secret shares, wherein each respective secret share of the at least one other plurality of secret shares is assigned to a respective SMPC compute engine of the plurality of SMPC compute engines.

9. The system of claim 8, wherein the second hardware processor of the respective SMPC compute engine is further configured to:

jointly, with the plurality of SMPC compute engines, hash the respective secret share from the plurality of secret shares with a secret salt value β and hash the respective secret share from the at least one other plurality of secret shares with the secret salt value β , where β is unknown to any SMPC compute engine of the plurality of SMPC compute engines;

transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

receive another plurality of hashed secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and

generate additional tokens, wherein each additional token is a combination of the plurality of hashed secret shares, irrespective of data source.

10. The system of claim 9, wherein the node and the at least one other node all receive the additional tokens.

11. The system of claim 7, wherein the data input is an identifier of a respective row or column in a dataset being uploaded by the node to the plurality of SMPC compute engines.

12. The system of claim 7, wherein the second hardware processor of the respective SMPC compute engine is further configured to:

- securely encrypt the respective secret share using an encryption scheme, wherein an initialization vector and key of the encryption scheme are in a secret share and are unknown to any SMPC compute engine of the plurality of SMPC compute engines;

- transmit the respective encrypted secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

- receive a plurality of encrypted secret shares from the remaining SMPC compute engines of the plurality of SMPC compute engines; and

- generate the token, wherein the token is a combination of the encrypted respective secret share and the plurality of encrypted secret shares.

13. A non-transitory computer readable medium storing thereon computer executable instructions for token generation using secure multiparty computation (SMPC) compute engines, comprising instructions for:

- hashing, by a node, a data input with a salt value;

- combining the hashed data input with a passcode provided by a user of the node;

- splitting, by the node, the hashed data input combined with the passcode into a plurality of secret shares, wherein each respective secret share of the plurality of secret shares is assigned to a respective SMPC compute engine of a plurality of SMPC compute engines,

- wherein the respective SMPC compute engine is configured to:

- securely hash the respective secret share with a secret salt value unique to the respective SMPC compute engine;

- transmit the respective hashed secret share to the remaining SMPC compute engines of the plurality of SMPC compute engines;

- receive a plurality of hashed secret shares from remaining SMPC compute engines of the plurality of SMPC compute engines;

- generate a token, wherein the token is a combination of the hashed respective secret share and the plurality of hashed secret shares;

- store a plurality of generated tokens with corresponding data inputs; and

in response to detecting that the stored plurality of generated tokens is being overwritten, generate an alert indicating malicious behaviour.

14. The non-transitory computer readable medium of claim 13, wherein the node is one data source of multiple data sources, wherein another token corresponding to data inputs from both the node and at least one other node of the multiple data sources should be generated, further comprising instructions for:

transmitting the salt value to the at least one other node, wherein the at least one other node is configured to:

hash at least one other data input with the salt value; and

split the hashed at least one other data input into at least one other plurality of secret shares, wherein each respective secret share of the at least one other plurality of secret shares is assigned to a respective SMPC compute engine of the plurality of SMPC compute engines.

15. The non-transitory computer readable medium of claim 13, wherein the data input is an identifier of a respective row or column in a dataset being uploaded by the node to the plurality of SMPC compute engines.