

Claims

1. 1. A computer-implemented method comprising:
 determining, at a participant of a plurality of participants, a function usable to map finite field elements, wherein the function is provided to the plurality of participants;
 generating, by each participant of the plurality of participants and based on a number of participants the function, polynomials evaluated in a set of points;
 distributing the polynomials, by each participant to corresponding each other participants of the plurality of participants, to determine a sum of the polynomials distributed to participants of the plurality of participants;
 determining a master polynomial from the sum of polynomials of each participant of distributed to the plurality of participants, wherein the ~~sum~~ master polynomial corresponds to a power of a secret; and
 -collectively generating, among the plurality of participants and based on the secret,
 a blockchain transaction.
2. 2. A ~~The computer-implemented~~ A method according to ~~any preceding claim 1,~~ wherein the set of points ~~are~~ is a set of elliptic curve points.
3. 3. The computer-implemented A ~~A~~ method according to ~~any preceding claim 1,~~ wherein determining the master polynomial from the sum of the polynomials is based on a Lagrange interpolation.
4. 4. The computer-implemented A ~~A~~ method according to ~~any preceding claim 1,~~ further comprising determining, based at least in part on the power of the secret, a common reference string that is determinable by the plurality of participants.

5. 5. The computer-implemented A-method according to ~~any preceding claim 1~~, further comprising receiving ~~corresponding~~ values of the polynomials from each participant of the plurality of participants, the values as evaluated in the set of points by each participant.
6. 6. The computer-implemented A-method according to ~~any preceding claim 5~~, wherein off-chain communications are utilised in ~~the~~ distribution of the values of the polynomials to each of the other ~~corresponding~~ participants.
7. The computer-implemented

7. ~~A method according to any preceding claim 1, wherein a result of the function as evaluated usable to map the finite field elements serves as a coefficients or exponent of a generator of a group.~~
8. ~~8. The computer-implemented A-method according to any preceding claim 1, wherein the plurality of participants is exactly two participants.~~
9. ~~9. The computer-implemented A-method according to any preceding claim 1, wherein the secret is determined based on the sum of master polynomials evaluated at a particular point.~~
10. ~~10. The computer-implemented A-method according to claim 9, wherein the particular point is an intersection point.~~
11. ~~11. The computer-implemented A-method according to any preceding claim 10, wherein the set of points excludes the intersection point.~~
12. ~~12. The computer-implemented A-method according to any preceding claim 10, wherein the polynomials evaluated in the set of points are of at least order 2.~~
13. ~~13. The computer-implemented A-method according to any preceding claim 1, wherein each point of the set of points is a different point.~~
14. ~~14. A system, comprising:~~
~~a processor; and~~
~~memory including executable instructions that, as a result of execution by the processor, causes the system to:~~ ~~perform the computer-implemented method according to any preceding claim.~~
~~determine, at a participant of plurality of participants, a function usable to map finite field elements, wherein the function is provided to the plurality of participants;~~
~~generate, by each participant of the plurality of participants and based on the function, polynomials evaluated in a set of points;~~

distribute the polynomials, by each participant to each other participant of the plurality of participants, to determine a sum of the polynomials distributed to the plurality of participants;
determine a master polynomial from the sum of the polynomials distributed to the plurality of participants, wherein the master polynomial corresponds to a power of a secret; and
collectively generate, among the plurality of participants and based on the secret, a blockchain transaction.

15. A non-transitory computer-readable storage medium having stored thereon executable instructions that, as a result of being executed by a processor of a computer system, cause the computer system to at least: ~~perform the computer-implemented method according to any of claims 1 to 13~~

determine, at a participant of plurality of participants, a function usable to map finite field elements, wherein the function is provided to the plurality of participants;

generate, by each participant of the plurality of participants and based on the function, polynomials evaluated in a set of points;

distribute the polynomials, by each participant to each other participant of the plurality of participants, to determine a sum of the polynomials distributed to the plurality of participants;

determine a master polynomial from the sum of the polynomials distributed to the plurality of participants, wherein the master polynomial corresponds to a power of a secret; and

collectively generate, among the plurality of participants and based on the secret, a blockchain transaction.

16. The system according to claim 14, wherein the set of points is a set of elliptic curve points.

17. The system according to claim 14, wherein determining the master polynomial from the sum of the polynomials is based on a Lagrange interpolation.

18. The system according to claim 14, wherein the executable instructions, as a result of execution by the processor, further cause the system to determine, based at least in part of the power of the secret, a common reference string that is determinable by the plurality of participants.
19. The non-transitory computer-readable storage medium according to claim 15, wherein the set of points is a set of elliptic curve points.
20. The non-transitory computer-readable storage medium according to claim 15, wherein determining the master polynomial from the sum of the polynomials is based on a Lagrange interpolation.

15. :