

Claims

1. A computer-implemented method comprising, at a verifying node of a blockchain network:
 - obtaining a first transaction which comprises runnable code;
 - receiving a second transaction which includes information comprising at least a submitted instance of an r-part and an s-part of a first ECDSA signature, and further comprising a nonce;
 - running the code from the first transaction, the code being configured to verify that $H_{PoW}(f(r, d))$ meets a predetermined condition defined in the code, and to return a result of true on condition thereof, where r is the submitted instance of the r-part, d is the nonce, H_{PoW} is a hash function, and f is a function combining r and d .
2. The method of claim 1, wherein f is a concatenation $r||d$.
3. The method of claim 1 or 2, wherein said predetermined condition is that $H_{puz}(r||d)$ is less than a predetermined target value, or that it has at least a predetermined minimum number of leading zeros.
4. The method of claim 1, 2 or 3, further comprising:
 - obtaining a public key wherein the first ECDSA signature signs a message based on a private key corresponding to the public key, the message being a part of the second transaction; and
 - applying an ECDSA verification function to verify the first ECDSA signature received in the second transaction based on the public key and the message, wherein the code is configured to return the result of true on further condition of said verification of the first ECDSA signature.
5. The method of any of claim 4, wherein the code further comprises a reference value corresponding to the r-part of the first ECDSA signature, the reference value being a reference instance of the r-part or a transformation thereof; and

the code is configured to check that the reference value corresponds to the reference instance of the r-part received in the second transaction, and return the result of true on further condition thereof.

6. The method of claim 5, wherein said reference value is a reference instance of the r-part of the ECDSA signature.

7. The method of claim 6, wherein the code is configured to perform:

check that $H_{PoW}(f(r, d))$ meets the predetermined condition, and
said ECDSA verification function performs:

compute $R' = H_{sig}(m)s^{-1} \cdot G + r's^{-1} \cdot P$, and

check that $[R']_x = r'$,

where r' is the reference instance of the r-part of the first ECDSA signature, s is the s-part of the first ECDSA signature, P is the first public key, m is the part of the second transaction signed by the first ECDSA signature, H_{sig} is a hash function that was used to hash m in generating the first ECDSA signature, G is an elliptic generator point, $[R']_x$ denotes an x-coordinate of R' , and “ $\cdot$ ” denotes an elliptic curve scalar multiplication; and

wherein the code is configured to return the result of true on condition that both said checks are true, but to return a result of false otherwise.

8. The method of claim 6, wherein the code is configured to perform:

check that $H_{PoW}(f(r, d))$ meets the predetermined condition, and

check that $r' = r$, and

wherein the ECDSA verification function performs:

compute $R' = H_{sig}(m)s^{-1} \cdot G + rs^{-1} \cdot P$, and

check that $[R']_x = r$,

where r is the submitted instance of the r-part of the first ECDSA signature, r' is the reference instance of the r-part of the first ECDSA signature, s is the s-part of the first ECDSA signature, P is the first public key, m is the part of the second transaction signed by the first ECC signature, H_{sig} is a hash function that was used to hash m in generating the first ECDSA

signature, G is an elliptic generator point, $[R']_x$ denotes an x-coordinate of R' , and “.” denotes an elliptic curve scalar multiplication; and

wherein the code is configured to return the result of true on condition that all three of said checks are true, but to return the result of false otherwise.

9. The method of claim 5, wherein said reference value is a transformation of a reference instance of the r-part of the first ECDSA signature, the code being configured to perform said check that the submitted instance corresponds to the reference value by: performing the same transformation on the submitted instance and comparing to the reference value.

10. The method of claim 9, wherein said reference value is a hash value, being a hash of the reference instance of the r-part of the first ECDSA signature.

11. The method of claim 10, wherein the code is configured to perform:

check that $H_{pow}(f(r, d))$ meets the predetermined condition, and

check that $h = H_{puz}(r)$, and

wherein the ECDSA verification function performs:

compute $R' = H_{sig}(m)s^{-1} \cdot G + rs^{-1} \cdot P$, and

check that $[R']_x = r$,

where r' is the reference instance of the r-part of the first ECDSA signature, r is the submitted instance of the r-part of the first ECDSA signature, s is the s-part of the first ECDSA signature, h is the hash value, H_{puz} is the hash function that was used to hash r' to generate h , P is the first public key, m is the part of the second transaction signed by the first ECC signature, H_{sig} is a hash function that was used to hash m in generating the first ECDSA signature, G is an elliptic generator point, $[R']_x$ denotes an x-coordinate of R' , and “.” denotes an elliptic curve scalar multiplication; and

wherein the code is configured to return the result of true on condition that all three of said checks are true, but to return a result of false otherwise.

12. The method of claim 4 or any claim dependent thereon, wherein said obtaining of the first public key comprises receiving the first public key as part of the information in the second transaction.

13. The method of any preceding claim, wherein the submitted instance of the r-part and s-part of the first ECDSA were generated by a second party using: an ephemeral key given to the second party by a first party or vice versa, and the first private key which is a private key of the second party; and

the nonce was also generated by the second party by performing a proof-of-work on computer equipment of the second party.

14. The method of claim 13, wherein:

$$P = V \cdot G,$$

$$k \in [1, n - 1],$$

$$R = k \cdot G,$$

$$r = [R]_x, \text{ and}$$

$$s = k^{-1}(H_{sig}(m) + rV) \bmod n,$$

where P is the first public key, V is the first private key, k is the ephemeral key, n is a prime modulus, G is an elliptic generator point, m is the part of the second transaction signed by the first ECDSA signature, H_{sig} is a hash function that was used to hash m in generating the first ECDSA signature, $[R]_x$ denotes an x-coordinate of R , and “ $\cdot$ ” denotes an elliptic curve scalar multiplication.

15. The method of claim 13 or 14, wherein said receiving of the second transaction comprises receiving the second transaction from the second party.

16. The method of any of claims 13 to 15, comprising triggering a service for the first party on condition of the result returned by said code being true.

17. The method of any of claims 13 to 16, wherein the information received in the second transaction comprises a further cryptographic signature of the second party signing

a part of the second transaction using a further private key of the second party, the further private key corresponding to a further public key.

18. The method of claim 17, wherein a mapping is available enabling the first party and/or a third party to look-up of an identity of the second party based on the further public key.

19. The method of claim 17 or 18, wherein the code is configured to verify the further cryptographic signature using the further public key and return the result of true on further condition that the further cryptographic signature is verified.

20. The method of any of claims 17 to 19, wherein the information received in the second transaction further comprises a cryptographic signature of the first party signing a part of the second transaction using a private key of the first party.

21. The method of any of claims 13 to 20, wherein:

the information received in the second transaction comprises an additional ECDSA signature having a different value of the r-part than the first ECDSA signature but using the same, first private key as the first ECDSA signature; and

the code is configured to verify the additional ECDSA signature using the first public key, and return the result of true on further condition that the additional ECDSA signature is verified.

22. The method of any preceding claim, wherein each of the transactions comprises a data structure comprising one or more inputs and one or more outputs, wherein each output comprises a locking script, and each input comprises an unlocking script and a pointer to an output of another transaction;

wherein said code is comprised by the locking script of the first transaction, wherein said information received in the second transaction is comprised by the unlocking script in an input of the second transaction, and wherein the pointer in said input of the second transaction points to said output of the first transaction; and

the method comprises validating the transaction at least on condition that the code returns said result of true, and in response to said validation, at least one of:

- including the second transaction in a pool of transactions for mining into one or more blocks by said verifying node, and/or
- forwarding the second transaction to at least one other of nodes of the blockchain network.

23. The method of any of claims 1 to 21, wherein the transactions are configured according to an account-based model, and said code is comprised by a smart contract included in the first transaction.

24. A computer program embodied on computer-readable storage and configured so as when run on a node of the network to perform the method of any preceding claim.

25. A node of the network, comprising:
 memory comprising one or more memory units, and
 processing apparatus comprising one or more processing units;
 wherein the memory stores code arranged to run on the processing apparatus, the code being configured so as when on the processing apparatus to carry out the method of any of claims 1 to 23.

26. A computer-implemented method comprising, at computer equipment of a second party:

- observing a first transaction which comprises runnable code, the code being configured to verify that $H_{PoW}(f(r, d))$ meets a predetermined condition defined in the code, and to return a result of true on condition thereof, where r is an r-part for an ECDSA signature, d is the nonce, H_{PoW} is a hash function, and f is a function combining r and d , the r-part being specified by a first party;
- generating the r-part based on an ephemeral key;
- searching for a value of the nonce d such that $H_{PoW}(f(r, d))$ meets the predetermined condition;

formulating a second transaction linked to the first transaction, the second transaction including information comprising at least the r-part and an s-part of a first ECDSA signature, and further comprising the nonce d ; and

sending the second transaction to be propagated over a blockchain network for recordal in a blockchain.

27. A set of transactions for recordal in a blockchain, the set comprising, embodied on a computer-readable medium or media:

a first transaction which comprises runnable code;

a second transaction which includes information comprising at least a submitted instance of an r-part and an s-part of a first ECDSA signature, and further comprising a nonce;

wherein the code is configured to verify that $H_{PoW}(f(r, d))$ meets a predetermined condition defined in the code, and to return a result of true on condition thereof, where r is the submitted instance of the r-part, d is the nonce, H_{PoW} is a hash function, and f is a function combining r and d .