CLAIMS

What is claimed is:

1. A method for block generation, comprising:

initiating, by a first consensus node of a blockchain, a target consensus proposal carrying a first timestamp provided by the first consensus node;

determining, by a second consensus node of the blockchain after receiving the target consensus proposal, a second timestamp based on a time that the target consensus proposal is received;

performing, by the second consensus node, verification on the first timestamp based on a third timestamp of a local ending block and the second timestamp to execute consensus logic on proposal data of the target consensus proposal after the verification succeeds; and

generating, by the first consensus node and the second consensus node if a consensus is reached on the target consensus proposal, a new block recording the proposal data of the target consensus proposal, wherein a fourth timestamp of the generated new block is obtained through determining based on at least the first timestamp, and is not earlier than timestamps of other local blocks of a consensus node to which the new block belongs;

wherein the blockchain adopts a consensus protocol based on a Honey Badger Byzantine Fault Tolerance algorithm, and a consensus stage to which the target consensus proposal belongs further comprises a consensus proposal initiated by the second consensus node, wherein the generated new block records proposal data of all consensus proposals on which consensuses are reached in the consensus stage, and the fourth timestamp is obtained through determining based on timestamps of all the consensus proposals on which consensuses are reached in the consensus stage, wherein the timestamps of all the consensus proposals comprise the first timestamp, and

wherein the fourth timestamp is a median value of the timestamps of all the consensus proposals on which consensuses are reached in the consensus stage.

2. The method according to claim 1, comprising:

the initiating, by a first consensus node of a blockchain, a target consensus proposal carrying a first timestamp provided by the first consensus node comprises:

obtaining, by the first consensus node of the blockchain when preparing to initiate the target consensus proposal, a fifth timestamp related to a current time and a sixth timestamp of a local ending block; and

if the fifth timestamp is later than the sixth timestamp, using, by the first consensus node, the fifth timestamp as the first timestamp, and initiating the target consensus proposal carrying the first timestamp; or

if the fifth timestamp is earlier than the sixth timestamp, determining, by the first consensus node, the first timestamp based on a preset rule, and initiating the target consensus proposal carrying the first timestamp, wherein the first timestamp determined based on the preset rule is obtained by adding a unit time to the sixth timestamp.

3. The method according to claim 1, wherein:

in performing verification on the first timestamp by the second consensus node of the blockchain, if the first timestamp is earlier than the second timestamp and later than the third timestamp, the verification succeeds; otherwise, the verification fails.

4. The method according to claim 1, wherein:

the second consensus node of the blockchain verifies the first timestamp based on the third timestamp of the local ending block and the second timestamp after a reliable broadcast (RBC) protocol stage of the Honey Badger Byzantine Fault Tolerance algorithm is completed and before a binary agreement (BA) protocol stage is entered.

5. The method according to claim 1, further comprising:

if the first timestamp is not successfully verified, the second consensus node sets a BA value of the target consensus proposal in a BA protocol stage to 0, to cause a failure to reach a consensus on the target consensus proposal.

6. A blockchain system comprising a plurality of consensus nodes configured to perform the method according to any one of the preceding claims.