CLAIMS

What is claimed is:

1. A secure multi-party computation method, comprising:

converting, by a first party, first plaintext data to the Montgomery state by using a first mapping operation, to obtain a first converted plaintext;

performing, by the first party, homomorphic encryption on the first converted plaintext in the Montgomery state to obtain the first converted ciphertext in the Montgomery state, and after performing the homomorphic encryption in the Montgomery state, directly outputting a result in an M ring as the first converted ciphertext, wherein the first mapping operation is used to convert data from an integer ring to the Montgomery state;

sending, by the first party, the first converted ciphertext to a second party; and

performing, by the second party, a first homomorphic operation in the Montgomery state based on the first converted ciphertext to obtain a first result ciphertext in the Montgomery state, wherein the first homomorphic operation comprises a modular multiplication operation, and wherein, during the first homomorphic operation, data is kept in the Montgomery state throughout a ciphertext operation stage so that a homomorphic ciphertext operation is continued in the Montgomery state.

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

sending, by the second party, the first result ciphertext to the first party; and

performing, by the first party, Montgomery reduction and a decryption operation on the first result ciphertext to obtain a first result plaintext.

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

sending, by the second party, the first result ciphertext to a third party; and

performing, by the third party, a second homomorphic operation in the Montgomery state based on the first result ciphertext to obtain a second result ciphertext.

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

obtaining, by the second party, a second converted ciphertext in the Montgomery state; and

the obtaining a first result ciphertext in the Montgomery state specifically comprises:

performing the first homomorphic operation on the first converted ciphertext and the second converted ciphertext to obtain the first result ciphertext.

5. The method according to claim 4, wherein the obtaining, by the second party, a second converted ciphertext in the Montgomery state specifically comprises:

receiving, by the second party, the second converted ciphertext from the first party.

6. The method according to claim 4, wherein the obtaining, by the second party, a second converted ciphertext in the Montgomery state specifically comprises:

reading, by the second party, local second plaintext data of the second party;

converting, by the second party, the local second plaintext data to the Montgomery state by using a first mapping operation, to obtain a second converted plaintext; and

performing, by the second party, homomorphic encryption on the second converted plaintext in the Montgomery state to obtain the second converted ciphertext, wherein the second party obtains the second converted ciphertext in a similar method in which the first party obtains the first converted ciphertext.

7. The method according to claim 6, wherein the first plaintext data is parameter data of a service prediction model, and the second plaintext data is characteristic data of a service object.

8. A secure multi-party computation method, wherein the method is executed by a first party and comprises:

converting first plaintext data to the Montgomery state by using a first mapping operation, to obtain a first converted plaintext;

performing homomorphic encryption on the first converted plaintext in the

Montgomery state to obtain the first converted ciphertext in the Montgomery state, and after performing the homomorphic encryption in the Montgomery state, directly outputting a result in an M ring as the first converted ciphertext, wherein the first mapping operation is used to convert data from an integer ring to the Montgomery state;

sending the first converted ciphertext to a second party;

receiving a result ciphertext from a third party, wherein the result ciphertext is obtained by performing a homomorphic operation in the Montgomery state based on the first converted ciphertext, and the homomorphic operation comprises a modular multiplication operation, and wherein, during the homomorphic operation, data is kept in the Montgomery state throughout a ciphertext operation stage so that a homomorphic ciphertext operation is continued in the Montgomery state; and

performing Montgomery reduction and a decryption operation on the result ciphertext to obtain a result plaintext.

9. The method according to claim 8, wherein the second party and the third party are the same party.

10. A secure multi-party computation method, wherein the method is executed by a second party and comprises:

receiving a first converted ciphertext in a Montgomery state from a first party, wherein the first converted ciphertext is obtained by the first party by converting the first plaintext data to the Montgomery state by using a first mapping operation, to obtain a first converted plaintext, and performing homomorphic encryption on the first converted plaintext in the Montgomery state to obtain the first converted ciphertext in the Montgomery state, wherein the first mapping operation is used to convert data from an integer ring to the Montgomery state, and wherein the first converted ciphertext is directly output by the first party as a result in an M ring after the homomorphic encryption is performed in the Montgomery state;

performing a first homomorphic operation in the Montgomery state based on the first converted ciphertext to obtain a first result ciphertext in the Montgomery state, wherein the first homomorphic operation comprises a modular multiplication operation, and wherein, during the first homomorphic operation, data is kept in the Montgomery

state throughout a ciphertext operation stage so that a homomorphic ciphertext operation is continued in the Montgomery state; and

sending the first result ciphertext.

11. The method according to claim 10, wherein the sending the first result ciphertext comprises:

sending the first result ciphertext to the first party, so that the first party performs Montgomery reduction and a decryption operation on the first result ciphertext to obtain a first result plaintext.

12. The method according to claim 10, wherein the sending the first result ciphertext comprises:

sending the first result ciphertext to a third party, so that the third party performs a second homomorphic operation in the Montgomery state based on the first result ciphertext to obtain a second result ciphertext.

13. The method according to claim 10, further comprising:

obtaining a second converted ciphertext in the Montgomery state; and

the obtaining a first result ciphertext in the Montgomery state specifically comprises:

performing the first homomorphic operation on the first converted ciphertext and the second converted ciphertext to obtain the first result ciphertext.

14. The method according to claim 13, wherein the obtaining a second converted ciphertext in the Montgomery state specifically comprises:

receiving the second converted ciphertext from the first party.

15. The method according to claim 13, wherein the obtaining a second converted ciphertext in the Montgomery state specifically comprises:

reading local second plaintext data;

converting the local second plaintext data to the Montgomery state by using the first mapping operation, to obtain a second converted plaintext; and

performing homomorphic encryption on the second converted plaintext in the Montgomery state to obtain the second converted ciphertext, wherein the second

converted ciphertext is obtained in a similar method in which the first party obtains the first converted ciphertext.

16. The method according to claim 15, wherein the first plaintext data is parameter data of a service prediction model, and the second plaintext data is characteristic data of a service object.

17. A secure multi-party computation system, comprising a first party and a second party, wherein

the first party is configured to convert first plaintext data to the Montgomery state by using a first mapping operation to obtain a first converted plaintext, perform homomorphic encryption on the first converted plaintext in the Montgomery state to obtain a first converted ciphertext in a Montgomery state, and after performing the homomorphic encryption in the Montgomery state, directly output a result in an M ring as the first converted ciphertext in the Montgomery state, wherein the first mapping operation is used to convert data from an integer ring to the Montgomery state;

the first party is further configured to send the first converted ciphertext to a second party; and

the second party is configured to perform a first homomorphic operation in the Montgomery state based on the first converted ciphertext to obtain a first result ciphertext in the Montgomery state, wherein the first homomorphic operation comprises a modular multiplication operation, and wherein, during the first homomorphic operation, data is kept in the Montgomery state throughout a ciphertext operation stage so that a homomorphic ciphertext operation is continued in the Montgomery state.

18. A secure multi-party computation apparatus, wherein the apparatus is deployed in a first party and comprises:

an encryption conversion unit, configured to convert first plaintext data to the Montgomery state by using a first mapping operation, to obtain a first converted plaintext, perform homomorphic encryption on the first converted plaintext in the Montgomery state to obtain a first converted ciphertext in the Montgomery state, and after performing the homomorphic encryption in the Montgomery state, directly output

a result in an M ring as the first converted ciphertext, wherein the first mapping operation is used to convert data from an integer ring to the Montgomery state;

a sending unit, configured to send the first converted ciphertext to a second party;

a receiving unit, configured to receive a result ciphertext from a third party, wherein the result ciphertext is obtained by performing a homomorphic operation in the Montgomery state based on the first converted ciphertext, and the homomorphic operation comprises a modular multiplication operation, and wherein, during the homomorphic operation, data is kept in the Montgomery state throughout a ciphertext operation stage so that a homomorphic ciphertext operation is continued in the Montgomery state; and

a decryption conversion unit, configured to perform Montgomery reduction and a decryption operation on the result ciphertext to obtain a result plaintext.

19. A secure multi-party computation apparatus, wherein the apparatus is deployed in a second party and comprises:

a receiving unit, configured to receive a first converted ciphertext in a Montgomery state from a first party, wherein the first converted ciphertext is obtained by the first party by converting the first plaintext data to the Montgomery state by using a first mapping operation, to obtain a first converted plaintext, and performing homomorphic encryption on the first converted plaintext in the Montgomery state to obtain the first converted ciphertext in the Montgomery state, wherein the first mapping operation is used to convert data from an integer ring to the Montgomery state, and wherein the first converted ciphertext is directly output by the first party as a result in an M ring after the homomorphic encryption is performed in the Montgomery state;

an operation unit, configured to perform a first homomorphic operation in the Montgomery state based on the first converted ciphertext to obtain a first result ciphertext in the Montgomery state, wherein the first homomorphic operation comprises a modular multiplication operation, and wherein, during the first homomorphic operation, data is kept in the Montgomery state throughout a ciphertext operation stage so that a homomorphic ciphertext operation is continued in the Montgomery state;

a sending unit, configured to send the first result ciphertext.

20. A computation device, comprising a memory and a processor, wherein the memory stores executable code, and the processor implements the method according to any one of claims 1 to 16 when executing the executable code.