METHOD, SYSTEM, AND DEVICE FOR SECURE DATA INTERACTION, AND STORAGE MEDIUM

FIELD

[0001] Embodiments of the present disclosure relate to the technical field of encrypted communications, and in particular to a method, a system and a device for secure data interaction, and a storage medium.

BACKGROUND

[0002] Private set intersection, or PSI for short, allows two parties holding respective data sets to compute an intersection of the data sets. At the end of the PSI, one or both of the two parties may acquire the intersection. However, neither party reveals anything to the data set of the counterparty except for elements in the intersection. The PSI is widely used in scenarios such as vertical federated learning.

[0003] The existing OPRF-based algorithm for the PSI fails to meet security requirements for a server when being maliciously attacked. The main reason for this is that in an oblivious transfer protocol, a client can break an encryption matrix of the server by violating the protocol to perform a malicious attack, for example, by entering a same protocol message in the oblivious transfer protocol, and thus obtain all the data of the server. Therefore, the above algorithm fails to meet the security requirements of data at the server when being maliciously attacked.

SUMMARY

[0004] In view of this, a method, a system and a device for secure data interaction and a storage medium are provided according to embodiments of the present disclosure, to solve the problem that the existing OPRF-based algorithm for the PSI fails to perform security verification during data transmission.

[0005] The following technical solutions are provided according to the embodiments of the present disclosure in order to solve this problem.

[0006] A method for secure data interaction is provided according to a first aspect of the embodiments of the present disclosure. The method includes: setting a public parameter at both a client and a server, and agreeing on a public function by the client and the server; performing a data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; and each time the data transmission protocol is performed, inputting a protocol message at the client, calculating a commitment value of the protocol message, and transmitting the protocol message and the commitment value of the protocol message to the server; after the data transmission protocol is performed the predetermined number of times, partially selecting protocol messages at the server randomly, opening the selected protocol messages by the client and transmitting the opened protocol messages by the client to the server, and performing security verification on the opened protocol messages by the server to obtain a security verification result; and performing private set intersection based on an unopened protocol message by the client and the server in a case of successful security verification, and acquiring, by the client, an intersection of respective data of the client and the server in the case of successful security verification.

[0007] Further, the public parameter includes a key security parameter λ, a first protocol parameter m, a second protocol parameter ω, a protocol security parameter s, a first length parameter l_1 and a second length parameter l_2, where $s=2\omega$. The public function includes a first hash function H_1, a second hash function H_2, a pseudo-random function F and a commitment function *Comm.*

[0008] Further, the method includes: randomly generating a selection string cs by the server based on the protocol security parameter s before the performing a data transmission protocol a predetermined number of times between the client and the server.

[0009] Further, the method includes: before the performing a data transmission protocol a predetermined number of times between the client and the server,

generating an initial matrix D by the client based on the first protocol parameter m and the protocol security parameter s, where the initial matrix D is a binary matrix with m rows and s columns, and all elements in the initial matrix D each are 1;

generating a binary key string k by random sampling based on the key security parameter

λ at the client;

calculating, for each element y in local data of the client, an encryption result v based on the first hash function H_1, the pseudo-random function F and the key string k from an equation:

$$v = F_k\left(H_1(y)\right)$$

where F_k represents the pseudo-random function F based on the key string k;

setting an element $D_i[v[i]]$ in the initial matrix D to 0 to obtain an updated ciphertext data matrix D', where $v[i]$ represents an i-th character in the encryption result v, $D_i[v[i]]$ represents an element in an i-th column and a $v[i]$-th row in the initial matrix D, and i is an integer greater than or equal to 0 and less than or equal to s;

randomly generating a first encryption matrix A with m rows and s columns by the client; and

calculating a second encryption matrix B based on the first encryption matrix A and the data matrix D' at the client from an equation:

$$B = A \oplus D'$$

where $\oplus$ represents exclusive OR of matrices.

[0010] Further, the performing a data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; and each time the data transmission protocol is performed, inputting a protocol message at the client, calculating a commitment value of the protocol message, and transmitting the protocol message and the commitment value of the protocol message to the server includes:

randomly generating a first commitment vector *ra* and a second commitment vector *rb* by the server based on the protocol security parameter s, and transmitting the first commitment vector *ra* and the second commitment vector *rb* by the server to the client;

each time the data transmission protocol is performed, acquiring a to-be-selected data group $\{Ai, Bi\}$ based on the first encryption matrix A and the second encryption matrix B by the client, as the protocol message, where A_i represents an i-th column of the first encryption matrix A, and B_i represents an i-th column of the second encryption matrix B;

calculating a first commitment calculation result $ca[i]$ and a second commitment calculation result $cb[i]$ by the client based on the to-be-selected data group $\{A_i, B_i\}$, the commitment function *Comm*, the first commitment vector *ra* and the second commitment vector *rb*, where the first commitment calculation result $ca[i]$ is calculated from an equation

$$ca[i] = Comm(A_i, ra[i]),$$

where $ra[i]$ represents an i-th element of the first commitment vector *ra*, and the second commitment calculation result $cb[i]$ is calculated from an equation

$$cb[i] = Comm(B_i, rb[i]),$$

where $rb[i]$ represents an i-th element of the second commitment vector *rb*; and

transmitting, by the client, the to-be-selected data group $\{A_i, B_i\}$, the first commitment calculation result $ca[i]$ and the second commitment calculation result $cb[i]$ to the server.

[0011] Further, the partially selecting protocol messages by the server randomly, opening the selected protocol messages by the client and transmitting the opened protocol messages by the client to the server, and performing security verification on the opened protocol messages by the server to obtain a security verification result includes:

selecting an element from the to-be-selected data group $\{A_i, B_i\}$ based on the selection string *cs*;

selecting an i-th column A_i of the first encryption matrix A as selected data c_i in a case that an i-th character $cs[i]$ in the selection string *cs* is 0;

selecting an i-th column B_i of the second encryption matrix B as the selected data c_i in a case that the i-th character $cs[i]$ in the selection string *cs* is 1;

acquiring a third encryption matrix C based on all the selected data c_i;

randomly generating a verification vector *ro* by the server based on the first protocol parameter ω, and transmitting the verification vector *ro* by the server to the client, where elements in the verification vector *ro* are each a positive integer less than or equal to *s*, and the elements in the verification vector *ro* are different from each other;

extracting, by the client, first verification data $A_{ro[j]}$ from the first encryption matrix A and

second verification data $B_{ro[j]}$ from the second encryption matrix B based on the verification vector ro, and returning the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ by the client to the server, where $ro[j]$ represents a j-th element of the verification vector ro, $A_{ro[j]}$ represents a $ro[j]$-th column of the first encryption matrix A, $B_{ro[j]}$ represents a $ro[j]$-th column of the second encryption matrix B, and j is an integer greater than or equal to 0 and less than or equal to ω;

calculating, by the server, a first commitment verification result da and a second commitment verification result db based on the first verification data $A_{ro[j]}$, the second verification data $B_{ro[j]}$, the commitment function $Comm$, the first commitment vector ra and the second commitment vector rb, where the first commitment verification result da is calculated from an equation:

$$da = Comm\left(A_{ro[j]}, ra\left[ro[j]\right]\right),$$

where $ra[ro[j]]$ represents a $ro[j]$-th element of the first commitment vector ra, the second commitment verification result db is calculated from an equation:

$$db = Comm\left(B_{ro[j]}, rb\left[ro[j]\right]\right),$$

where $rb[ro[j]]$ represents a $ro[j]$-th element of the second commitment vector rb;

performing, by the server, security verification on the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ based on the first commitment calculation result $ca[i]$, the second commitment calculation result $cb[i]$, the first commitment verification result da, the second commitment verification result db and the third encryption matrix C;

determining whether all of a first condition, a second condition and a third condition are met, where the first condition is that the first verification data $A_{ro[j]}$ is different from the second verification data $B_{ro[j]}$; the second condition is that a first commitment calculation result $ca[ro[j]]$ corresponding to the j-th element of the verification vector ro is equal to the first commitment verification result da, and a second commitment calculation result $cb[ro[j]]$ corresponding to the j-th element of the verification vector ro is equal to the second commitment verification result db; and the third condition is that a $ro[j]$-th column $C_{ro[j]}$ of the third encryption matrix C is equal to the first verification data $A_{ro[j]}$ in a case that a $ro[j]$-th character $cs[ro[j]]$ of the selection string cs is 0, and the $ro[j]$-th column $C_{ro[j]}$ of the third

encryption matrix C is equal to the second verification data $B_{ro[j]}$ in a case that the $ro[j]$-th character $cs[ro[j]]$ of the selection string cs is 1;

determining that the security verification is unsuccessful in a case that one of the first condition, the second condition and the third condition is not met, and ending data interaction between the client and the server; and

determining that the security verification is successful in a case that all of the first condition, the second condition and the third condition are met.

[0012] Further, the performing private set intersection based on an unopened protocol message by the client and the server, and acquiring, by the client, the intersection of respective data of the client and the server includes:

deleting, by the client, the first verification data $A_{ro[j]}$ from the first encryption matrix A, to obtain a fourth encryption matrix E with m rows and ω columns, and transmitting the key string k by the client to the server;

deleting, by the server, the $ro[j]$-th column $C_{ro[j]}$ of the third encryption matrix C from the third encryption matrix C, to obtain a fifth encryption matrix G with m rows and ω columns;

calculating, by the server for each element x in local data of the server, a first encryption vector p based on the first hash function H_1, the pseudo-random function F and the key string k, from an equation:

$$p = F_k\left(H_1(x)\right)$$

deleting, by the server, $p[ro[j]]$ from the first encryption vector p, to obtain a second encryption vector p', where $p[ro[j]]$ represents a $ro[j]$-th character of the first encryption vector p

calculating, by the server, a first encryption ciphertext φ_1 based on the second encryption vector p', the fifth encryption matrix G and the second hash function H_2 from an equation:

$$\varphi_1 = H_2(G_1\left[p'[1]\right] \| \cdots \| G_\omega\left[p'[\omega]\right]);$$

where $p'[\omega]$ represents a ω-th character of the second encryption vector p', $G_\omega[p'[\omega]]$ represents an element in a ω-th column and $p'[\omega]$-th row of the fifth encryption matrix G;

transmitting the first encryption ciphertext φ_1 by the server to the client;

suppose server has *srv_m* elements in local data, server sends *srv_m* φ_{1S} of all its local elements, φ_1[i] corresponds the i-th element of server.

calculating, by the client for each element y in the local data of the client, a third encryption vector q based on the first hash function H_1, the pseudo-random function F and the key string k from an equation:

$$q = F_k\big(H_1(y)\big) ;$$

deleting, by the client, $q[ro[j]]$ from the third encryption vector q, to obtain a fourth encryption vector q', where $q[ro[j]]$ represents a $ro[j]$-th character of the third encryption vector q;

calculating, by the client, a second encryption ciphertext φ_2 based on the fourth encryption vector q', the fourth encryption matrix E and the second hash function H_2 from an equation:

$$\varphi_2 = H_2(E_1\left[q'[1]\right] \| \cdots \| E_\omega\left[q'[\omega]\right])$$

where $q'[\omega]$ represents a $ro[j]$-th character of the fourth encryption vector q', $E_\omega[q'[\omega]]$ represents an element in a ω-th column and a $q'[\omega]$-th row of the fourth encryption matrix E; and

comparing the second encryption ciphertext φ_2 with the first encryption ciphertext φ_1 by the client, and acquiring, by the client, an intersection of the local data of the server and the local data of the client.

suppose client has *cli_m* elements in local data, client calculates *cli_m* φ_{2S} of all its local elements, φ_2[i] corresponds the i-th element of client. And client recieves *srv_m* φ_{1S} of server elements. Client compares each of its φ_2 with each φ_1 of server elements to find identical pairs which mean they are in intersection result.

[0013] A system for secure data interaction is provided according to a second aspect of the

embodiments of the present disclosure. The system includes a client and a server, and is configured: to set a public parameter at both the client and the server, and for the client and the server agree on a public function; to perform a data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; and each time the data transmission protocol is performed, to input a protocol message at the client, calculate a commitment value of the protocol message, and to transmit the protocol message and the commitment value of the protocol message to the server; after the data transmission protocol is performed the predetermined number of times, to partially select protocol messages at the server randomly, to open the selected protocol messages at the client and transmitting the opened protocol messages from the client to the server, and to perform security verification on the opened protocol messages at the server to obtain a security verification result; and to perform private set intersection based on an unopened protocol message at the client and the server in a case of successful security verification, and to acquire, at the client, an intersection of respective data of the client and the server in the case of successful security verification.

[0014] Further, the public parameter includes a key security parameter λ, a first protocol parameter m, a second protocol parameter ω, a protocol security parameter s, a first length parameter l_1 and a second length parameter l_2, where $s=2\omega$. The public function includes a first hash function H_1, a second hash function H_2, a pseudo-random function F and a commitment function *Comm.*

[0015] Further, the system is configured to: randomly generating a selection string cs at the server based on the protocol security parameter s before the performing a data transmission protocol a predetermined number of times between the client and the server.

[0016] Further, the system is configured to: before perform a data transmission protocol a predetermined number of times between the client and the server,

generate an initial matrix D at the client based on the first protocol parameter m and the protocol security parameter s, where the initial matrix D is a binary matrix with m rows and s columns, and all elements in the initial matrix D each are 1;

generate a binary key string k by random sampling based on the key security parameter λ at the client;

calculate, for each element y in local data of the client, an encryption result v based on the first hash function H_1, the pseudo-random function F and the key string k from an equation:

$$v = F_k\left(H_1(y)\right)$$

where F_k represents the pseudo-random function F based on the key string k;

set an element $D_i[v[i]]$ in the initial matrix D to 0 to obtain an updated ciphertext data matrix D', where $v[i]$ represents an i-th character in the encryption result v, $D_i[v[i]]$ represents an element in an i-th column and a $v[i]$-th row in the initial matrix D, and i is an integer greater than or equal to 0 and less than or equal to s;

randomly generate a first encryption matrix A with m rows and s columns at the client; and

calculate a second encryption matrix B based on the first encryption matrix A and the data matrix D' at the client from an equation:

$$B = A \oplus D'$$

where $\oplus$ represents exclusive OR of matrices.

[0017] Further, in order to the perform the data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; and each time the data transmission protocol is performed, input a protocol message at the client, calculate a commitment value of the protocol message, and transmit the protocol message and the commitment value of the protocol message to the server, the system is configured to:

randomly generate a first commitment vector *ra* and a second commitment vector *rb* at the server based on the protocol security parameter s, and transmit the first commitment vector *ra* and the second commitment vector *rb* to the client;

each time the data transmission protocol is performed, acquire a to-be-selected data group $\{Ai, Bi\}$ based on the first encryption matrix A and the second encryption matrix B at the client, as the protocol message, where A_i represents an i-th column of the first encryption matrix A, and B_i represents an i-th column of the second encryption matrix B;

calculate a first commitment calculation result $ca[i]$ and a second commitment calculation result $cb[i]$ at the client based on the to-be-selected data group $\{A_i, B_i\}$, the commitment function *Comm*, the first commitment vector *ra* and the second commitment vector *rb*, where the first commitment calculation result $ca[i]$ is calculated from an equation

$$ca[i] = Comm(A_i, ra[i]),$$

where $ra[i]$ represents an i-th element of the first commitment vector *ra*, and the second commitment calculation result $cb[i]$ is calculated from an equation

$$cb[i] = Comm(B_i, rb[i]),$$

where $rb[i]$ represents an i-th element of the second commitment vector *rb*; and

transmit the to-be-selected data group $\{A_i, B_i\}$, the first commitment calculation result $ca[i]$ and the second commitment calculation result $cb[i]$ from the client to the server.

[0018] Further, in order to partially select protocol messages by the server randomly, open the selected protocol messages by the client and transmit the opened protocol messages from the client to the server, and perform security verification on the opened protocol messages at the server to obtain a security verification result, the system is configured to:

select an element from the to-be-selected data group $\{A_i, B_i\}$ based on the selection string *cs*;

select an i-th column A_i of the first encryption matrix A as selected data c_i in a case that an i-th character $cs[i]$ in the selection string *cs* is 0;

select an i-th column B_i of the second encryption matrix B as the selected data c_i in a case that the i-th character $cs[i]$ in the selection string *cs* is 1;

acquire a third encryption matrix C based on all the selected data c_i;

randomly generate a verification vector *ro* by the server based on the first protocol parameter ω, and transmit the verification vector *ro* by the server to the client, where elements in the verification vector *ro* are each a positive integer less than or equal to s, and the elements in the verification vector *ro* are different from each other;

extract, by the client, first verification data $A_{ro[j]}$ from the first encryption matrix A and

second verification data $B_{ro[j]}$ from the second encryption matrix B based on the verification vector ro, and returning the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ by the client to the server, where $ro[j]$ represents a j-th element of the verification vector ro, $A_{ro[j]}$ represents a $ro[j]$-th column of the first encryption matrix A, $B_{ro[j]}$ represents a $ro[j]$-th column of the second encryption matrix B, and j is an integer greater than or equal to 0 and less than or equal to ω;

calculate, by the server, a first commitment verification result da and a second commitment verification result db based on the first verification data $A_{ro[j]}$, the second verification data $B_{ro[j]}$, the commitment function $Comm$, the first commitment vector ra and the second commitment vector rb, where the first commitment verification result da is calculated from an equation:

$$da = Comm\left(A_{ro[j]}, ra\left[ro[j]\right]\right),$$

where $ra[ro[j]]$ represents a $ro[j]$-th element of the first commitment vector ra, the second commitment verification result db is calculated from an equation:

$$db = Comm\left(B_{ro[j]}, rb\left[ro[j]\right]\right),$$

where $rb[ro[j]]$ represents a $ro[j]$-th element of the second commitment vector rb;

perform, by the server, security verification on the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ based on the first commitment calculation result $ca[i]$, the second commitment calculation result $cb[i]$, the first commitment verification result da, the second commitment verification result db and the third encryption matrix C;

determine whether all of a first condition, a second condition and a third condition are met, where the first condition is that the first verification data $A_{ro[j]}$ is different from the second verification data $B_{ro[j]}$; the second condition is that a first commitment calculation result $ca[ro[j]]$ corresponding to the j-th element of the verification vector ro is equal to the first commitment verification result da, and a second commitment calculation result $cb[ro[j]]$ corresponding to the j-th element of the verification vector ro is equal to the second commitment verification result db; and the third condition is that a $ro[j]$-th column $C_{ro[j]}$ of the third encryption matrix C is equal to the first verification data $A_{ro[j]}$ in a case that a $ro[j]$-th character $cs[ro[j]]$ of the selection string cs is 0, and the $ro[j]$-th column $C_{ro[j]}$ of the third

encryption matrix C is equal to the second verification data $B_{ro[j]}$ in a case that the $ro[j]$-th character $cs[ro[j]]$ of the selection string cs is 1;

determine that the security verification is unsuccessful in a case that one of the first condition, the second condition and the third condition is not met, and ending data interaction between the client and the server; and

determine that the security verification is successful in a case that all of the first condition, the second condition and the third condition are met.

[0019] Further, in order to perform private set intersection based on an unopened protocol message by the client and the server, and acquiring, by the client, the intersection of respective data of the client and the server, the system is configured to:

delete, at the client, the first verification data $A_{ro[j]}$ from the first encryption matrix A, to obtain a fourth encryption matrix E with m rows and ω columns, and transmitting the key string k by the client to the server;

delete, at the server, the $ro[j]$-th column $C_{ro[j]}$ of the third encryption matrix C from the third encryption matrix C, to obtain a fifth encryption matrix G with m rows and ω columns;

calculate, at the server for each element x in local data of the server, a first encryption vector p based on the first hash function H_1, the pseudo-random function F and the key string k, from an equation:

$$p = F_k\left(H_1(x)\right)$$

delete, at the server, $p[ro[j]]$ from the first encryption vector p, to obtain a second encryption vector p', where $p[ro[j]]$ represents a $ro[j]$-th character of the first encryption vector p

calculate, at the server, a first encryption ciphertext φ_1 based on the second encryption vector p', the fifth encryption matrix G and the second hash function H_2 from an equation:

$$\varphi_1 = H_2(G_1\left[p'[1]\right] \| \cdots \| G_\omega\left[p'[\omega]\right]);$$

where $p'[\omega]$ represents a ω-th character of the second encryption vector p', $G_\omega[p'[\omega]]$ represents an element in a ω-th column and $p'[\omega]$-th row of the fifth encryption matrix G;

transmit the first encryption ciphertext φ_1 from the server to the client;

calculate, at the client for each element y in the local data of the client, a third encryption vector q based on the first hash function H_1, the pseudo-random function F and the key string k from an equation:

$$q = F_k\left(H_1(y)\right);$$

delete, at the client, $q[ro[j]]$ from the third encryption vector q, to obtain a fourth encryption vector q', where $q[ro[j]]$ represents a $ro[j]$-th character of the third encryption vector q;

calculate, at the client, a second encryption ciphertext φ_2 based on the fourth encryption vector q', the fourth encryption matrix E and the second hash function H_2 from an equation:

$$\varphi_2 = H_2(E_1\left[q'[1]\right] \| \cdots \| E_\omega\left[q'[\omega]\right])$$

where $q'[\omega]$ represents a $ro[j]$-th character of the fourth encryption vector q', $E_\omega[q'[\omega]]$ represents an element in a ω-th column and a $q'[\omega]$-th row of the fourth encryption matrix E; and

compare the second encryption ciphertext φ_2 with the first encryption ciphertext φ_1 at the client, and acquire, at the client, an intersection of the local data of the server and the local data of the client.

[0020] A device for secure data interaction is provided according to a third aspect of the embodiments of the present disclosure. The device includes a processor and a memory. The memory is configured to store one or more program instructions. The processor is configured to execute the one or more program instructions to perform the method for secure data interaction according to the first aspect of the present disclosure as described above.

[0021] A computer-readable storage medium is provided according to a fourth aspect of the embodiments of the present disclosure. The computer-readable storage medium stores a computer program. The computer program, when being executed by a processor, implements the method for secure data interaction according to the first aspect of the present disclosure as described above.

[0022] The present disclosure has the following advantages. A method, a system and a

device for secure data interaction, as well as a storage medium are provided. First, the public parameter is set at both the client and the server, and the client and the server agree on the public function. Then, the data transmission protocol is performed a predetermined number of times between the client and the server. Each time the data transmission protocol is performed, the protocol message is inputted at the client, and a commitment value of the protocol message is calculated. The protocol message and the commitment value are transmitted to the server. After the data transmission protocol, some protocol messages are randomly selected at the server. The client opens the selected protocol messages and transmit the opened selected protocol messages to the server. The server performs security verification on the opened protocol messages. Private set intersection is performed by the client and the server in a case of successful security verification, and the client acquires an intersection of respective data of the client and the server. According to the embodiments of the present disclosure, security verification is added to the data transmission protocol so as to ensure security of data of the server under malicious attacks, thereby preventing the client from acquiring the data of the server via the malicious attacks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] In order to illustrate technical solutions of the embodiments of the present disclosure or in the conventional technology more clearly, the drawings used in the description of the embodiments or the conventional technology are briefly described below. Apparently, the drawings in the following description are only illustrative. Other drawings may be obtained by those skilled in the art from the drawings without any creative work.

[0024] The structure, the scale, the size and the like shown in this specification are only used to match the contents disclosed in this specification for those skilled in the art to understand and read, rather than limit the implementations of the present disclosure, and thus has no technical significance. Any modification in the structure, change in the scale or modification in the size, without affecting the technical effects and the purpose the present disclosure is to achieve, should fall within the scope of the technical contents disclosed in the present disclosure.

[0025] Figure 1 is a schematic structural diagram illustrating a system for secure data interaction according to an embodiment of the present disclosure;

[0026] Figure 2 is a schematic flow chart illustrating a method for secure data interaction according to an embodiment of the present disclosure;

[0027] Figure 3 is a schematic flow chart illustrating data pre-processing according to an embodiment of the present disclosure;

[0028] Figure 4 is a schematic flow chart illustrating a data transmission protocol according to an embodiment of the present disclosure;

[0029] Figure 5 is a schematic flow chart illustrating security verification according to an embodiment of the present disclosure; and

[0030] Figure 6 is a schematic flow chart illustrating private set intersection according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

[0031] The embodiments of the present disclosure are illustrated below by the specific examples. Those skilled in the art may easily understand the other advantages and effects of the present disclosure from the contents disclosed in this specification. It is apparent that the described embodiments are only some rather than all of the embodiments of the present disclosure. All other embodiments obtained by those skilled in the art based on the embodiments of the present disclosure without any creative work fall within the protection scope of the present disclosure.

[0032] Reference is made to Figure 1. A system for secure data interaction is provided according to an embodiment of the present disclosure. The system includes a client and a server.

[0033] Further, the system is configured: to set a public parameter at the client and the server and for the client and the server to agree on a public function; perform a data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; each time the data transmission protocol is performed, input a protocol message at the client, calculate a commitment value of the protocol message, and transmit the protocol message and the commitment value to the server; randomly select some protocol messages at the server, and open the selected protocol messages at the client and transmit the opened protocol message from the client to the server,

perform security verification on the opened protocol messages at the server to acquire a security verification result; and perform private set intersection based on an unopened protocol message between the client and the server, and acquire an intersection of respective data of the client and the server at the client.

[0034] The system for secure data interaction is provided according to the embodiment of the present disclosure. First, the public parameter is set at both the client and the server, and the client and the server agree on the public function. Then, the data transmission protocol is performed a predetermined number of times between the client and the server. Each time the data transmission protocol is performed, the protocol message is inputted at the client, and a commitment value of the protocol message is calculated. The protocol message and the commitment value are transmitted to the server. After the data transmission protocol, some protocol messages are randomly selected at the server. The client opens the selected protocol messages and transmit the opened selected protocol messages to the server. The server performs security verification on the opened protocol messages. Private set intersection is performed by the client and the server in a case of successful security verification, and the client acquires an intersection of respective data of the client and the server. According to the embodiments of the present disclosure, security verification is added to the data transmission protocol so as to ensure security of data of the server under malicious attacks, thereby preventing the client from acquiring the data of the server via the malicious attacks.

[0035] Corresponding to the system for secure data interaction disclosed above, a method for secure data interaction is further provided according to an embodiment of the present disclosure. The method for secure data interaction according to the embodiment of the present disclosure is described in detail below with reference to the system for secure data interaction described above.

[0036] As shown in Figure 2, the method for secure data interaction according to the embodiment of the present disclosure is described below.

[0037] A public parameter is set at both the client and the server, and the client and the server agree on a public function, which are described as follows. The public parameter is set and the client and the server agree on the public function based on respective data volumes of the client and the server. The public parameter includes: a key security parameter λ, a first protocol parameter m, a second protocol parameter ω, a protocol security parameter s, a first

length parameter l_1 and a second length parameter l_2, where s=2ω. For example, a maximum number of elements in data of each of the client and the server is 10 million, the first protocol parameter m is set to 2^{20}, the second protocol parameter is set to ω=2^{10}, the protocol security parameter s is set to 2^{21}, the first length parameter l_1 is set to 256, and the second length parameter l_2 is set as 128.

[0038] The public function includes a first hash function H_1, a second hash function H_2 and a pseudo-random function F. The first hash function H_1 is expressed as follows.

$$H_1: \{0,1\}^* \rightarrow \{0,1\}^{l_1}$$

where $\{0,1\}^* \rightarrow \{0,1\}^{l_1}$ represents conversion from $\{0,1\}^*$ into $\{0,1\}^{l_1}$, $\{0,1\}^*$ represents a binary string with a length of any value, and $\{0,1\}^{l_1}$ represents a binary string with a length of l_1.

[0039] The second hash function H_2 is expressed as follows.

$$H_2: \{0,1\}^{\omega} \rightarrow \{0,1\}^{l_2}$$

where $\{0,1\}^{\omega} \rightarrow \{0,1\}^{l_2}$ represents conversion from a binary string with a length of ω into a binary string with a length of l_2.

[0040] The pseudo-random function F is expressed as follows.

$$F: \{0,1\}^{\lambda} \times \{0,1\}^{l_1} \rightarrow [m]^s$$

where $\{0,1\}^{\lambda} \times \{0,1\}^{l_1} \rightarrow [m]^s$ represents conversion from a binary string with a length of λ and a binary string with a length of l_1 into a string formed by s integers less than m.

[0041] A commitment function *Comm* is expressed as: t=*Comm*(n,r). n represents a protocol message provided by the client, and r represents a random number provided by the server. The commitment function is in various forms. For example, a *Hash* summary function may serve as the commitment function, that is, *Comm*(n,r)=*SHA*256 ($n||r$), where $n||r$ represents concatenation of n and r.

[0042] As shown in Figure 3, data pre-processing is performed at the server and the client before the data transmission protocol is performed the predetermined number of times between the client and the server.

[0043] The data pre-processing is performed as follows. A selection string *cs* is randomly generated by the server based on the protocol security parameter *s*. The selection string *cs* is a random binary string with a length of *s*. An initial matrix *D* is generated at the client based on the first protocol parameter *m* and the protocol security parameter *s*. The initial matrix *D* is a binary matrix with *m* rows and *s* columns, and all elements in the initial matrix *D* are 1. A binary key string *k* is generated by random sampling based on the key security parameter λ at the client. For each element *y* in local data of the client, an encryption result *v* is calculated based on the first hash function H_1, the pseudo-random function *F* and the key string *k*. The encryption result *v* is calculated from the following equation: $v = F_k(H_1(y))$. F_{k} represents a pseudo-random function *F* based on the key string *k*, and the encryption result *v* is a string formed by *s* integers less than *m*. An element $D_i[v[i]]$ in the initial matrix *D* is set to 0. The initial matrix *D* is updated based on each element *y* in the local data of the client, to obtain an updated ciphertext data matrix *D*'. $v[i]$ represents an *i*-th character in the encryption result *v*. $D_i[v[i]]$ represents an element in an *i*-th column and a $v[i]$-th row in the initial matrix *D*, and *i* is an integer greater than or equal to 0 and less than or equal to *s*. A first encryption matrix *A* with *m* rows and *s* columns is randomly generated at the client. A second encryption matrix *B* is calculated based on the first encryption matrix *A*, and the data matrix *D*' of the client. The second encryption matrix *B* is expressed as: $B = A \oplus D'$, where $\oplus$ represents exclusive OR of matrices.

[0044] The data transmission protocol is performed *s* times between the client and the server based on the public parameter and the public function. Each time the data transmission protocol is performed, the protocol message is inputted at the client, the commitment value of the protocol message is calculated, and the protocol message and the commitment value of the protocol message are transmitted to the server.

[0045] Referring to Figure 4, this processing is described in detail as follows. A first commitment vector *ra* and a second commitment vector *rb* are randomly generated at the server based on the protocol security parameter *s*. The first commitment vector *ra* and the second commitment vector *rb* are transmitted to the client. Each of the first commitment vector *ra* and the second commitment vector *rb* is formed by *s* random numbers. Each time the data transmission protocol is performed, a to-be-selected data group $\{A_i, B_i\}$ is acquired based on the first encryption matrix *A* and the second encryption matrix *B* at the client, as the protocol message. A_i represents an *i*-th column of the first encryption matrix *A*, and B_i represents an *i*-th column of the second encryption matrix *B*. The client calculates a first commitment calculation result *ca*[*i*] and a second commitment calculation result *cb*[*i*] based on the to-be-selected data group $\{A_i, B_i\}$, the commitment function *Comm*, the first commitment vector *ra* and the second commitment vector *rb*. The first commitment calculation result *ca*[*i*] is calculated from the following equation: $ca[i] = Comm(A_i, ra[i])$. *ra*[*i*] represents an *i*-th element of the first commitment vector *ra*. The second commitment calculation result *cb*[*i*] is calculated from an equation: $cb[i] = Comm(B_i, rb[i])$. *rb*[*i*] represents an *i*-th element of the second commitment vector *rb*. The client transmits the to-be-selected data group $\{A_i, B_i\}$, the first commitment calculation result *ca*[*i*] and the second commitment calculation result *cb*[*i*] to the server.

[0046] After the data transmission protocol is performed *s* times, the server randomly selects some protocol messages. The client opens the selected protocol messages and transmits the opened protocol messages to the server. The server performs security verification on the opened protocol messages, to acquire the security verification result.

[0047] Referring to Figure 5, this processing is described in detail as follows. The server selects an element from the to-be-selected data group $\{A_i, B_i\}$ based on the selection string *cs*. An *i*-th column A_i of the first encryption matrix *A* is selected as selected data c_i in a case that an *i*-th character *cs*[*i*] in the selected string *cs* is 0. An *i*-th column B_i of the second encryption matrix *B* is selected as the selected data c_i in a case that the *i*-th character *cs*[*i*] in the selected string *cs* is 1. A third encryption matrix *C* is acquired based on all the selected data c_i. The

server randomly generates a verification vector *ro* based on the first protocol parameter ω, and transmits the verification vector *ro* to the client. Elements in the verification vector *ro* are each a positive integer less than or equal to *s*. Elements in the verification vector *ro* are different from each other. The client extracts first verification data $A_{ro[j]}$ from the first encryption matrix *A* and second verification data $B_{ro[j]}$ from the second encryption matrix *B* based on the verification vector *ro*, and returns the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ to the server. *ro*[*j*] represents a *j*-th element of the verification vector *ro*. $A_{ro[j]}$ represents a *ro*[*j*]-th column of the first encryption matrix *A*. $B_{ro[j]}$ represents a *ro*[j]-th column of the second encryption matrix *B*. *j* represents an integer greater than or equal to 0 and less than or equal to ω. The server calculates a first commitment verification result *da* and a second commitment verification result *db* based on the first verification data $A_{ro[j]}$, the second verification data $B_{ro[j]}$, the commitment function *Comm*, the first commitment vector *ra* and the second commitment vector *rb*. The first commitment verification result *da* is calculated from the following equation:

$$da = Comm\left(A_{ro[j]}, ra\left[ro[j]\right]\right) .$$

ra[*ro*[*j*]] represents a *ro*[*j*]-th element of the first commitment vector *ra*. The second commitment verification result *db* is calculated from the following equation:

$$db = Comm\left(B_{ro[j]}, rb\left[ro[j]\right]\right)$$

rb[*ro*[*j*]] represents a *ro*[*j*]-th element of the second commitment vector *rb*.

The server security performs security verification on the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ based on the first commitment calculation result *ca*[*i*], the second commitment calculation result *cb*[*i*], the first commitment verification result *da*, the second commitment verification result *db* and the third encryption matrix *C*. It is determined whether both the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ meet a first condition, a second condition and a third condition. The first condition is that the first verification data $A_{ro[j]}$ is different from the second verification data $B_{ro[j]}$. The second condition is that a first commitment calculation result *ca*[*ro*[*j*]] corresponding to the *j*-th element of the verification vector *ro* is equal to the first commitment verification result *da*, and a second

commitment calculation result $cb[ro[j]]$ corresponding to the j-th element of the verification vector ro is equal to the second commitment verification result db. The third condition is that a $ro[j]$-th column $C_{ro[j]}$ of the third encryption matrix C is equal to the first verification data $A_{ro[j]}$ in a case that a $ro[j]$-th character $cs[ro[j]]$ of the selection string cs is 0, and the $ro[j]$-th column $C_{ro[j]}$ of the third encryption matrix C is equal to the second verification data $B_{ro[j]}$ in a case that the $ro[j]$-th character $cs[ro[j]]$ of the selection string cs is 1. The security verification is unsuccessful in a case that one of the first condition, the second condition and the third condition is not met, and the data interaction between the client and the server ends. The security verification is successful in a case that all of the first condition, the second condition and the third condition are met.

[0048] Private set intersection is performed between the client and the server based on an unopened protocol message, and the client acquires the intersection of respective data of the client and the server.

[0049] Referring to Figure 6, this processing is described in detail below. The client deletes the first verification data $A_{ro[j]}$ from the first encryption matrix A, to obtain a fourth encryption matrix E with m rows and ω columns, and transmits the key string k to the server. The server deletes the $ro[j]$-th column $C_{ro[j]}$ of the third encryption matrix C from the third encryption matrix C, to obtain a fifth encryption matrix G with m rows and ω columns. The server, for each element x in local data of the server, calculates a first encryption vector p based on the first hash function H_1, the pseudo-random function F and the key string k. The first encryption vector p is calculated from the following equation:

$$p = F_k\left(H_1(x)\right) .$$

The server deletes $p[ro[j]]$ from the first encryption vector p, to obtain a second encryption vector p'. $p[ro[j]]$ represents a $ro[j]$-th character of the first encryption vector p. The server calculates a first encryption ciphertext φ_1 based on the second encryption vector p', the fifth encryption matrix G and the second hash function H_2. The first encryption ciphertext φ_1 is calculated from the following equation;

$$\varphi_1 = H_2(G_1\left[p'[1]\right] \| \cdots \| G_\omega\left[p'[\omega]\right])$$

$p'[\omega]$ represents a ω-th character of the second encryption vector p'. $G_\omega[p'[\omega]]$ represents

an element in a ω-th column and $p'[\omega]$-th row of the fifth encryption matrix G. The server transmits the first encryption ciphertext φ_1 to the client.

The client, for each element y in the local data of the client, calculates a third encryption vector q based on the first hash function H_1, the pseudo-random function F and the key string k. The third encryption vector q is calculated from the following equation:

$$q = F_k\big(H_1(y)\big) ;$$

The client deletes $q[ro[j]]$ from the third encryption vector q, to obtain a fourth encryption vector q'. $q[ro[j]]$ represents a $ro[j]$-th character of the third encryption vector q. The client calculates a second encryption ciphertext φ_2 based on the fourth encryption vector q', the fourth encryption matrix E and the second hash function H_2. The second encryption ciphertext φ_2 is calculated from the following equation:

$$\varphi_2 = H_2(E_1\left[q'[1]\right] \| \cdots \| E_\omega\left[q'[\omega]\right])$$

$q'[\omega]$ represents a $ro[j]$-th character of the fourth encryption vector q', $E_\omega[q'[\omega]]$ represents an element in a ω-th column and a $q'[\omega]$-th row of the fourth encryption matrix E.

[0050] The client compares the second encryption ciphertext φ_2 with the received first encryption ciphertext φ_1, and acquires an intersection of the local data of the server and the local data of the client.

[0051] The method for secure data interaction is provided according to the embodiment of the present disclose. First, the public parameter is set at both the client and the server, and the client and the server agree on the public function. Then, the data transmission protocol is performed a predetermined number of times between the client and the server. Each time the data transmission protocol is performed, the protocol message is inputted at the client, and a commitment value of the protocol message is calculated. The protocol message and the commitment value are transmitted to the server. After the data transmission protocol, some protocol messages are randomly selected at the server. The client opens the selected protocol messages and transmit the opened selected protocol messages to the server. The server performs security verification on the opened protocol messages. Private set intersection is performed by the client and the server in a case of successful security verification, and the

client acquires an intersection of respective data of the client and the server. According to the embodiments of the present disclosure, security verification is added to the data transmission protocol so as to ensure security of data of the server under malicious attacks, thereby preventing the client from acquiring the data of the server via the malicious attacks.

[0052] In addition, a device for secure data interaction is further provided according to an embodiment of the present disclosure. The device includes a processor and a memory. The memory is configured to store one or more program instructions. The processor is configured to execute the one or more program instructions to perform the method for secure data interaction according to any embodiment as described above.

[0053] In addition, a computer-readable storage medium is further provided according to an embodiment of the present disclosure. The computer-readable storage medium stores a computer program. The computer program, when being executed by a processor, implements the method for secure data interaction according to any embodiment as described above.

[0054] In the embodiments of the present disclosure, the processor may be an integrated circuit chip capable of processing a signal. The processor may be a general-purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), other programmable logic device, a discrete gate, a transistor logic device, or a discrete hardware component.

[0055] The processor may implement or perform the method, the steps and logical block diagrams in the embodiments of the present disclosure. The general-purpose processor may be a microprocessor or any conventional processor. The steps of the method according to the embodiments of the present disclosure may be directly performed by a hardware decoder, or may be performed by a hardware module together with a software module in the decoder. The software module may be located in the existing storage medium in the art, for example, a random memory, a flash memory, a read-only memory, a programmable read-only memory, an electrically erasable programmable memory, or a register. The processor reads information in the storage medium and performs the steps of the above method together with the hardware of the processor.

[0056] The storage medium may be a memory, for example, a volatile memory or a non-volatile memory, or may include both the volatile memory and the non-volatile memory.

[0057] The non-volatile memory may be a read-only memory (ROM), a programmable ROM (PROM), an erasable PROM (EPROM), an electrically EPROM (EEPROM) or a flash memory.

[0058] The volatile memory may be a random-access memory (RAM), and serves as an external cache. By way of illustration rather than limitation, various RAMs such as a static RAM (SRAM), a dynamic RAM (DRAM), a synchronous DRAM (SDRAM), a double data rate SDRAM (DDRSDRAM), an enhanced SDRAM (ESDRAM), a synchlink DRAM (SLDRAM) and a direct Rambus RAM (DRRAM) are applicable.

[0059] The storage medium in the embodiments of the present disclosure is intended to include, but is not limited to the listed memories and any other suitable memories.

[0060] Those skilled in the art should realize that the functions described in the present disclosure may be implemented by both hardware and software in one or more examples described above. In a case that the functions are implemented by software, the functions may be stored in a computer-readable medium or transmitted as one or more instructions or codes in the computer-readable medium. The computer-readable medium includes a computer storage medium and a communication medium. The communication medium includes any medium that facilitates transmission of a computer program from one place to another. The storage medium may be any available medium accessible to a general-purpose computer or a specific computer.

[0061] Although the present disclosure is described in detail above with general descriptions and specific embodiments, some modifications or improvements may be made to the present disclosure, which is apparent to those skilled in the art. Therefore, these modifications or improvements made without departing from the spirit of the present disclosure fall within the protection scope of the present disclosure.

CLAIMS

1. A method for secure data interaction, comprising:

setting a public parameter at both a client and a server, and agreeing on a public function by the client and the server;

performing a data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; and each time the data transmission protocol is performed, inputting a protocol message at the client, calculating a commitment value of the protocol message, and transmitting the protocol message and the commitment value of the protocol message to the server;

after the data transmission protocol is performed the predetermined number of times, partially selecting protocol messages at the server randomly, opening the selected protocol messages by the client and transmitting the opened protocol messages by the client to the server, and performing security verification on the opened protocol messages by the server to obtain a security verification result; and

performing private set intersection based on an unopened protocol message by the client and the server in a case of successful security verification, and acquiring, by the client, an intersection of respective data of the client and the server in the case of successful security verification.

2. The method for secure data interaction according to claim 1, wherein

the public parameter comprises a key security parameter λ, a first protocol parameter m, a second protocol parameter ω, a protocol security parameter s, a first length parameter l_1 and a second length parameter l_2, wherein $s=2\omega$; and

the public function comprises a first hash function H_1, a second hash function H_2, a pseudo-random function F and a commitment function *Comm.*

3. The method for secure data interaction according to claim 2, further comprising:

randomly generating a selection string cs by the server based on the protocol security parameter s before the performing a data transmission protocol a predetermined number of times between the client and the server.

4. The method for secure data interaction according to claim 3, further comprising:

before the performing a data transmission protocol a predetermined number of times between the client and the server,

generating an initial matrix D by the client based on the first protocol parameter m and the protocol security parameter s, wherein the initial matrix D is a binary matrix with m rows and s columns, and all elements in the initial matrix D each are 1;

generating a binary key string k by random sampling based on the key security parameter λ at the client;

calculating, for each element y in local data of the client, an encryption result v based on the first hash function H_1, the pseudo-random function F and the key string k from an equation:

$$v = F_k\left(H_1(y)\right)$$

wherein F_k represents the pseudo-random function F based on the key string k;

setting an element $D_i[v[i]]$ in the initial matrix D to 0 to obtain an updated ciphertext data matrix D', wherein $v[i]$ represents an i-th character in the encryption result v, $D_i[v[i]]$ represents an element in an i-th column and a $v[i]$-th row in the initial matrix D, and i is an integer greater than or equal to 0 and less than or equal to s;

randomly generating a first encryption matrix A with m rows and s columns at the client; and

calculating a second encryption matrix B based on the first encryption matrix A and the data matrix D' by the client from an equation:

$$B = A \oplus D'$$

wherein $\oplus$ represents exclusive OR of matrices.

5. The method for secure data interaction according to claim 4, wherein the performing a data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; and each time the data transmission protocol is performed, inputting a protocol message at the client, calculating a commitment

value of the protocol message, and transmitting the protocol message and the commitment value of the protocol message to the server comprises:

randomly generating a first commitment vector *ra* and a second commitment vector *rb* by the server based on the protocol security parameter *s*, and transmitting the first commitment vector *ra* and the second commitment vector *rb* to the client;

each time the data transmission protocol is performed, acquiring a to-be-selected data group {*Ai*, *Bi*} based on the first encryption matrix *A* and the second encryption matrix *B* by the client, as the protocol message, wherein A_i represents an *i*-th column of the first encryption matrix *A*, and B_i represents an *i*-th column of the second encryption matrix *B*;

calculating a first commitment calculation result *ca*[*i*] and a second commitment calculation result *cb*[*i*] by the client based on the to-be-selected data group {A_i, B_i}, the commitment function *Comm*, the first commitment vector *ra* and the second commitment vector *rb*, wherein the first commitment calculation result *ca*[*i*] is calculated from an equation

$$ca[i] = Comm(A_i, ra[i]),$$

wherein *ra*[*i*] represents an *i*-th element of the first commitment vector *ra*, and the second commitment calculation result *cb*[*i*] is calculated from an equation

$$cb[i] = Comm(B_i, rb[i]),$$

wherein *rb*[*i*] represents an *i*-th element of the second commitment vector *rb*; and

transmitting, by the client, the to-be-selected data group {A_i, B_i}, the first commitment calculation result *ca*[*i*] and the second commitment calculation result *cb*[*i*] to the server.

6. The method for secure data interaction according to claim 5, wherein the partially selecting protocol messages at the server randomly, opening the selected protocol messages by the client and transmitting the opened protocol messages by the client to the server, and performing security verification on the opened protocol messages by the server to obtain a security verification result comprises:

selecting an element from the to-be-selected data group {A_i, B_i} based on the selection string *cs*;

selecting an *i*-th column A_i of the first encryption matrix *A* as selected data c_i in a case

that an i-th character $cs[i]$ in the selection string cs is 0;

selecting an i-th column B_i of the second encryption matrix B as the selection data c_i in a case that the i-th character $cs[i]$ in the selected string cs is 1;

acquiring a third encryption matrix C based on all the selected data c_i;

randomly generating a verification vector ro by the server based on the first protocol parameter ω, and transmitting the verification vector ro to the client, wherein elements in the verification vector ro are each a positive integer less than or equal to s, and the elements in the verification vector ro are different from each other;

extracting, by the client, first verification data $A_{ro[j]}$ from the first encryption matrix A and second verification data $B_{ro[j]}$ from the second encryption matrix B based on the verification vector ro, and returning the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ to the server, wherein $ro[j]$ represents a j-th element of the verification vector ro, $A_{ro[j]}$ represents a $ro[j]$-th column of the first encryption matrix A, $B_{ro[j]}$ represents a $ro[j]$-th column of the second encryption matrix B, and j is an integer greater than or equal to 0 and less than or equal to ω;

calculating, by the server, a first commitment verification result da and a second commitment verification result db based on the first verification data $A_{ro[j]}$, the second verification data $B_{ro[j]}$, the commitment function $Comm$, the first commitment vector ra and the second commitment vector rb, wherein the first commitment verification result da is calculated from an equation:

$$da = Comm\left(A_{ro[j]}, ra\left[ro[j]\right]\right),$$

wherein $ra[ro[j]]$ represents a $ro[j]$-th element of the first commitment vector ra, the second commitment verification result db is calculated from an equation:

$$db = Comm\left(B_{ro[j]}, rb\left[ro[j]\right]\right),$$

wherein $rb[ro[j]]$ represents a $ro[j]$-th element of the second commitment vector rb;

performing, by the server, security verification on the first verification data $A_{ro[j]}$ and the second verification data $B_{ro[j]}$ based on the first commitment calculation result $ca[i]$, the second commitment calculation result $cb[i]$, the first commitment verification result da, the

second commitment verification result *db* and the third encryption matrix *C*;

determining whether all of a first condition, a second condition and a third condition are met, wherein the first condition is that the first verification data $A_{ro[j]}$ is different from the second verification data $B_{ro[j]}$; the second condition is that a first commitment calculation result *ca*[*ro*[*j*]] corresponding to the *j*-th element of the verification vector *ro* is equal to the first commitment verification result *da*, and a second commitment calculation result *cb*[*ro*[*j*]] corresponding to the *j*-th element of the verification vector *ro* is equal to the second commitment verification result *db*; and the third condition is that a *ro*[*j*]-th column $C_{ro[j]}$ of the third encryption matrix *C* is equal to the first verification data $A_{ro[j]}$ in a case that a *ro*[*j*]-th character *cs*[*ro*[*j*]] of the selection string *cs* is 0, and the *ro*[*j*]-th column $C_{ro[j]}$ of the third encryption matrix *C* is equal to the second verification data $B_{ro[j]}$ in a case that the *ro*[*j*]-th character *cs*[*ro*[*j*]] of the selection string *cs* is 1;

determining that the security verification is unsuccessful in a case that one of the first condition, the second condition and the third condition is not met, and ending data interaction between the client and the server; and

determining that the security verification is successful in a case that all of the first condition, the second condition and the third condition are met.

7. The method for secure data interaction according to claim 6, wherein the performing private set intersection based on an unopened protocol message by the client and the server in, and acquiring, by the client, the intersection of respective data of the client and the server comprises:

deleting, by the client, the first verification data $A_{ro[j]}$ from the first encryption matrix *A*, to obtain a fourth encryption matrix *E* with *m* rows and ω columns, and transmitting the key string *k* by the client to the server;

deleting, by the server, the *ro*[*j*]-th column $C_{ro[j]}$ of the third encryption matrix *C* from the third encryption matrix *C*, to obtain a fifth encryption matrix *G* with *m* rows and ω columns;

calculating, by the server for each element *x* in local data of the server, a first encryption vector *p* based on the first hash function H_1, the pseudo-random function *F* and the key string *k*, wherein the first encryption vector *p* is calculated from an equation:

$$p = F_k\left(H_1(x)\right)$$

deleting, by the server, $p[ro[j]]$ from the first encryption vector p, to obtain a second encryption vector p', wherein $p[ro[j]]$ represents a $ro[j]$-th character of the first encryption vector p;

calculating, by the server, a first encryption ciphertext φ_1 based on the second encryption vector p', the fifth encryption matrix G and the second hash function H_2, wherein the first encryption ciphertext φ_1 is calculated from an equation:

$$\varphi_1 = H_2(G_1\left[p'[1]\right] \| \cdots \| G_\omega\left[p'[\omega]\right]);$$

wherein $p'[\omega]$ represents a ω-th character of the second encryption vector p', $G_\omega[p'[\omega]]$ represents an element in a ω-th column and $p'[\omega]$-th row of the fifth encryption matrix G;

transmitting the first encryption ciphertext φ_1 by the server to the client;

calculating, by the client, for each element y in the local data of the client, a third encryption vector q based on the first hash function H_1, the pseudo-random function F and the key string k from an equation:

$$q = F_k\left(H_1(y)\right);$$

deleting, by the client, $q[ro[j]]$ from the third encryption vector q, to obtain a fourth encryption vector q', wherein $q[ro[j]]$ represents a $ro[j]$-th character of the third encryption vector q;

calculating, by the client, a second encryption ciphertext φ_2 based on the fourth encryption vector q', the fourth encryption matrix E and the second hash function H_2 from an equation:

$$\varphi_2 = H_2(E_1\left[q'[1]\right] \| \cdots \| E_\omega\left[q'[\omega]\right])$$

wherein $q'[\omega]$ represents a $ro[j]$-th character of the fourth encryption vector q', $E_\omega[q'[\omega]]$ represents an element in a ω-th column and a $q'[\omega]$-th row of the fourth encryption matrix E; and

comparing the second encryption ciphertext φ_2 with the first encryption ciphertext φ_1 by the client, and acquiring, by the client, an intersection of the local data of the server and the local data of the client.

8. A system for secure data interaction, comprising: a client and a server, and the system is configured:

to set a public parameter at both the client and the server, and for the client and the server agree on a public function;

to perform a data transmission protocol a predetermined number of times between the client and the server based on the public parameter and the public function; and each time the data transmission protocol is performed, input a protocol message at the client, calculate a commitment value of the protocol message, and transmit the protocol message and the commitment value of the protocol message to the server;

after the data transmission protocol is performed the predetermined number of times, to partially select protocol messages at the server randomly, to open the selected protocol messages at the client and transmitting the opened protocol messages from the client to the server, and to perform security verification on the opened protocol messages at the server to obtain a security verification result; and

to perform private set intersection based on an unopened protocol message at the client and the server in a case of successful security verification, and to acquire, at the client, an intersection of respective data of the client and the server in the case of successful security verification.

9. A device for secure data interaction, comprising:

a memory configured to store one or more program instructions; and

a processor configured to execute the one or more program instructions to perform the method for secure data interaction according to any one of claims 1 to 7.

10. A computer-readable storage medium storing a computer program that, when being executed by a processor, implements the method for secure data interaction according to any one of claims 1 to 7.