

METHOD AND APPARATUS FOR UNBALANCED PRIVACY SET INTERSECTION

TECHNICAL FIELD

[0001] The present application relates to the field of computer application and information
5 technologies, and in particular, to a method and an apparatus for unbalanced PSI, a computer-
readable storage medium, and an electronic device.

BACKGROUND

[0002] With the enactment and improvement of laws and regulations such as Cybersecurity
Law and Personal Information Protection Law, data compliance and privacy security increasingly
10 attract more attention. Emerging privacy computing technologies provide key support for data
privacy security and for being available and unrecognizable. PSI is a very important algorithm in
current privacy computing, and has been applied in a large scale in a plurality of scenarios such as
government affairs, finance, and medical care. PSI (private set intersection) can also be referred
to as secure intersection, and is one of basic functions of secure multi-party computation (SMPC
15 or MPC for short). When participating parties hold their own private data sets (that is, private sets),
PSI can enable any party to obtain an intersection of the private data sets of the other parties while
protecting data privacy of each party.

[0003] However, most of current private data set intersection technologies are in a balanced
scenario, that is, amounts of data held by participants are approximately equal. In an unbalanced
20 scenario, that is, when a difference between amounts of data held by participants is large, a related
private data set intersection technology is still being developed and needs improvement. In the
existing technologies, some solutions implement secure PSI in an unbalanced scenario by using a
fully homomorphic encryption technology. However, there are some inherent disadvantages in
these solutions, for example, problems such as low ciphertext computing efficiency, a large
25 ciphertext size, and high complexity of an algorithm requires complex optimization
implementation.

SUMMARY

[0004] For the above disadvantages existing in a PSI method in an unbalanced scenario in the existing technologies, the present application provides a method and an apparatus for unbalanced PSI, a computer-readable storage medium, and an electronic device, so that ciphertext computing efficiency is higher and compatibility is better.

[0005] According to an aspect of the present application, a method for unbalance PSI is provided. A first party holds a first data set, a second party holds a second data set, and the method is performed by the first party and includes: performing data preprocessing on private data in the first data set to obtain a first mapped data set; obtaining a function in a polynomial form through fitting based on the first mapped data set; combining coefficients of all terms in the function into a coefficient vector; receiving a public key from the second party, and performing homomorphic encryption on the coefficient vector by using the public key to obtain an encrypted coefficient vector; receiving a ciphertext input vector from the second party, and obtaining a ciphertext result through computation with reference to the encrypted coefficient vector; and sending the ciphertext result to the second party, so that the second party obtains a result of unbalanced PSI.

[0006] In an implementation, the performing data preprocessing on the first data set further includes: performing data preprocessing by using OPRF, where all data in the first mapped data set are pseudorandom numbers.

[0007] In an implementation, the private data of the first party is a user identifier of a to-be-queried user, and private data of the second party is a group of user identifiers of a target category.

[0008] In an implementation, the ciphertext input vector is obtained based on following steps: generating a random matrix; generating an identity matrix; generating an invertible matrix pair; and obtaining the ciphertext input vector through computation based on the random matrix, the identity matrix, the invertible matrix pair, and a private key of the second party.

[0009] In an implementation, the function is obtained through fitting by using a least squares method/procedure.

[0010] According to an aspect of the present application, a method for unbalance PSI is provided. A first party holds a first data set, a second party holds a second data set, and the method is performed by the second party and includes: performing data preprocessing on private data in the second data set to obtain a second mapped data set; generating a pair of a private key and a

public key, and sending the public key to the first party; encrypting data in the second mapped data set based on the private key to generate a ciphertext input vector and a decryption key; sending the ciphertext input vector to the first party; and receiving a ciphertext result based on the ciphertext input vector from the first party, and decrypting the ciphertext result by using the
5 decryption key to obtain a result of unbalanced PSI.

[0011] In an implementation, the preprocessing is performed by using OPRF, and all data in the second mapped data set are pseudorandom numbers.

[0012] In an implementation, the ciphertext input vector is obtained based on following steps: generating a random matrix; generating an identity matrix; generating an invertible matrix pair;
10 and obtaining the ciphertext input vector through computation based on the private key, the random matrix, the identity matrix, and the invertible matrix pair.

[0013] In an implementation, the decryption key is generated based on following steps: generating a random matrix; generating an identity matrix; generating an invertible matrix pair; and generating the decryption key through computation based on the random matrix, the identity
15 matrix, and the invertible matrix pair.

[0014] In an implementation, if the result of the unbalanced PSI obtained through the decrypting is 0, private data of the first party is intersection data; and if the result of the unbalanced PSI obtained through the decrypting is not 0, the private data of the first party is not the intersection data.

20 **[0015]** In an implementation, private data of the first party is a user identifier of a to-be-queried user, and the private data of the second party is a group of user identifiers of a target category.

[0016] According to an aspect of the present application, an apparatus for unbalanced PSI is provided. A first party holds a first data set, a second party holds a second data set, and the apparatus is disposed in the first party and includes: a preprocessing unit, configured to perform
25 data preprocessing on private data in the first data set to obtain a first mapped data set; a fitting unit, configured to: obtain a function in a polynomial form through fitting based on the first mapped data set, and combine coefficients of all terms in the function into a coefficient vector; an encryption unit, configured to: receive a public key from the second party, and perform homomorphic encryption on the coefficient vector by using the public key to obtain an encrypted
30 coefficient vector; a computation unit, configured to: receive a ciphertext input vector from the second party, and obtain a ciphertext result through computation with reference to the encrypted

coefficient vector; and a sending unit, configured to send the ciphertext result to the second party, so that the second party obtains a result of unbalanced PSI.

[0017] In an implementation, the preprocessing unit performs data preprocessing by using OPRF, and all data in the first mapped data set are pseudorandom numbers.

5 **[0018]** In an implementation, the function is obtained through fitting by using a least squares method/procedure.

[0019] In an implementation, the private data of the first party is a user identifier of a to-be-queried user, and private data of the second party is a group of user identifiers of a target category.

[0020] According to an aspect of the present application, an apparatus for unbalanced PSI is
10 provided. A first party holds a first data set, a second party holds a second data set, and the apparatus is disposed in the second party and includes: a preprocessing unit, configured to perform data preprocessing on private data in the second data set to obtain a second mapped data set; a key generation unit, configured to: generate a pair of a private key and a public key based on a homomorphic encryption algorithm, and send the public key to the first party; an encryption unit,
15 configured to encrypt data in the second mapped data set based on the private key to generate a ciphertext input vector and a decryption key; a sending unit, configured to send the ciphertext input vector to the first party; and a decryption unit, configured to: receive a ciphertext result based on the ciphertext input vector from the first party, and decrypt the ciphertext result by using the decryption key to obtain a result of unbalanced PSI.

20 **[0021]** In an implementation, the preprocessing unit performs data preprocessing by using OPRF, and all data in the second mapped data set are pseudorandom numbers.

[0022] In an implementation, the encryption unit obtains the ciphertext input vector based on following steps: generating a random matrix; generating an identity matrix; generating an invertible matrix pair; and obtaining the ciphertext input vector through computation based on the
25 private key, the random matrix, the identity matrix, and the invertible matrix pair.

[0023] In an implementation, the encryption unit generates the decryption key based on following steps: generating a random matrix; generating an identity matrix; generating an invertible matrix pair; and generating the decryption key through computation based on the random matrix, the identity matrix, and the invertible matrix pair.

30 **[0024]** In an implementation, if the result of the unbalanced PSI obtained through the decrypting is 0, it is determined that private data of the first party is intersection data; and if the

result of the unbalanced PSI obtained through the decrypting is not 0, the private data of the first party is not the intersection data.

[0025] In an implementation, private data of the first party is a user identifier of a to-be-queried user, and the private data of the second party is a group of user identifiers of a target category.

5 **[0026]** According to an aspect of the present application, a computer-readable storage medium is provided. The computer-readable storage medium stores instructions, and when the instructions are run on a processor, the method for unbalanced PSI according to an aspect of the present application is performed.

[0027] According to an aspect of the present application, an electronic device is provided,
10 including: one or more processors, and a memory. The memory stores one or more computer programs, and the one or more computer programs include instructions. When the instructions are executed by the processor, the processor is enabled to perform the method for unbalanced PSI according to an aspect of the present application.

[0028] It can be appreciated from the above that, in the implementations of the present
15 specification, the first party holding the first data set can obtain, from the second party holding the second data set, only the ciphertext input vector obtained through homomorphic encryption, and the first party cannot infer any data included in the second data set by using the ciphertext input vector. The first party sends the ciphertext result to the second party, and the ciphertext result is also obtained by the first party through linear transformation computation by using two encryption
20 matrices. The second party decrypts the ciphertext result by using the decryption key, and can only determine, based on the decrypted ciphertext result, whether the private data of the first party belongs to the second data set, and cannot inversely deduce and determine a specific location of to-be-queried data in the data set, so that a security requirement of an anonymous query can be satisfied. Compared with the existing technologies, in the method for unbalanced PSI in this
25 solution, a complex computation process of PSI is converted into a simple matrix transformation operation, which not only ensures ciphertext computing efficiency and retrieval efficiency, but also can implement many advantages such as batch processing and compatibility.

BRIEF DESCRIPTION OF DRAWINGS

[0029] The example implementations of the present application can be further appreciated

with reference to the accompanying drawings, to understand the aspects of the present application more clearly.

[0030] FIG. 1 is an example block diagram illustrating a procedure of a method for unbalanced PSI according to a first implementation of the present application;

5 **[0031]** FIG. 2 is an example block diagram illustrating a procedure of a method for unbalanced PSI according to a second implementation of the present application;

[0032] FIG. 3 is an example schematic diagram illustrating a system architecture of an apparatus for unbalanced PSI according to a third implementation of the present application;

[0033] FIG. 4 is an example schematic diagram illustrating a system architecture of an apparatus for unbalanced PSI according to a fourth implementation of the present application; and

10 **[0034]** FIG. 5 is an example diagram illustrating a structure of a computer-readable storage medium corresponding to a method for unbalanced PSI according to another implementation of the present application.

DESCRIPTION OF EMBODIMENTS

15 **[0035]** To make the technical content disclosed in the present application more detailed and complete, references can be made to the accompanying drawings and the following example implementations of the present application. Same reference numerals in the accompanying drawings represent same or similar components. However, a person of ordinary skill in the art should understand that the implementations provided below are not intended to limit the scope covered by the present application. In addition, the accompanying drawings are merely used for example description, and are not drawn based on original sizes of the accompanying drawings.

20 **[0036]** The following further describes in detail example implementations of each aspect of the present application with reference to the accompanying drawings.

[0037] The term used in one or more implementations of the present specification is merely used for the purpose of describing a particular implementation and is not intended to limit one or more implementations of the present specification. The singular forms "a", "the", and "the" used in one or more implementations of the present specification and the appended claims are also intended to include most forms unless the context clearly represents other meanings. It should be further understood that the term "and/or" used in one or more implementations of the present

specification refers to and includes any of or all possible combinations of one or more associated listed items. It should be understood that, although the terms such as "first" and "second" can be used in one or more implementations of the present specification to describe various types of information, the information should not be limited to these terms. These terms are merely used to distinguish information of a same type from each other. For example, without departing from the scope of one or more implementations of the present specification, "first" can be referred to as "second", and similarly, "second" can be referred to as "first". Depending on the context, the term "if" used herein can be used in an exchangeable manner with or be referred to as "when ..." or "during ..." or "in response to determining that ...".

10 **[0038]** The flowchart is used in the present specification to describe the operations performed by the system according to the implementations of the present specification. It should be understood that previous or subsequent operations are not necessarily performed precisely in an order. Instead, the steps can be processed in a reverse order or simultaneously. In addition, other operations can be added to these processes, or one operation or several operations can be removed from these processes.

[0039] First, terms in the description of one or more implementations of the present application are explained.

[0040] Private set intersection: This is referred to as PSI for short. Assuming that bank A has IDs of a group of users and financial institution B has IDs of a group of users, an intersection of user IDs of the two institutions can be identified by using PSI while data privacy is protected, and no information other than intersection data is disclosed. PSI is widely used in various scenarios such as finance, government affairs, and medical care, and is one of the most mature technologies in privacy computing.

25 **[0041]** Unbalanced PSI: It is assumed that a size of an ID data set of bank A is X and a size of an ID data set of financial institution B is Y. When $X \ll Y$, PSI in this scenario is referred to as the unbalanced PSI. If the existing PSI technology is directly used, bank A with a smaller data amount requires a same amount of computation and same communication overheads as institution B with a larger data amount.

30 **[0042]** Homomorphic encryption: This is a special public key encryption solution. An operation in a ciphertext domain obtained after decryption is equivalent to a corresponding operation in a plaintext domain.

[0043] Somewhat homomorphic encryption: This is a type of homomorphic encryption that supports both ciphertext addition and multiplication operations only with a limited multiplication depth.

[0044] Oblivious pseudorandom function: This is referred to as "OPRF" for short, and is a cryptographic protocol in which two participants Sender and Receiver exist, where the Sender provides a key k and a function F , and the Receiver provides an input x . When the cryptographic protocol is run, the Receiver obtains an output $F(k, x)$, and the Sender has no output. In a protocol running process, the Sender does not know the input x , and the Receiver does not know the function F and the key k .

[0045] Private data: This includes but is not limited to personal information (a personal name, birthday, nation, nationality, a family relationship, an address, a personal telephone number, an email, etc.), personal identity information (an identity card, a military officer certificate, a passport, a driving license, an employee's card, an access card, a social security card, a residence permit, etc.), personal biometric recognition information (personal genes, fingerprints, voiceprints, eyeprints, palmprints, auricles, irises, facial features, etc.), network identity identification information (a system account number, an IP address, an email address, a password and a password protection answer related to the above, a user personal digital certificate, etc.), personal health and physiological information (related records generated due to illness and treatment of an individual, such as a disease, an inpatient medical record, a doctor's order, a test report, a surgery and anesthesia record, a nursing record, a medication administration record, drug and food allergy information, birth information, a past medical history, a diagnosis and treatment condition, a family medical history, a current medical history, and an infectious disease history, as well as generated information related to a personal body health status, a weight, a height, a vital capacity, etc.), personal education and work information (a personal occupation and position, a workplace, an academic qualification, an academic degree, education experience, work experience, a training record, a transcript, etc.), personal property information (a bank account number, authentication information (a password), deposit information (including a fund amount, payment and collection records, etc., real estate information, a credit record, credit information, transaction and consumption records, a flow record, etc., as well as virtual property information such as virtual currency, virtual transactions, and game redemption codes), personal communication information (communication records and content, text messages, multimedia messages, emails, and data

(which is usually referred to as metadata) describing personal communication, etc.), contact information (an address book, a friend list, a group list, an email address list, etc.), personal Internet access records (refer to user operation records stored in logs, including website browsing records, software usage records, click records, etc.), personal frequently used device information (refers to
5 information describing a basic status of a personal frequently used device, including a hardware serial number, a device MAC address, a software list, a unique device identification code (such as an IMEI/android ID/IDFA/OPENUDID/GUID, or IMSI information of a SIM card.), personal location information (including a track route, precise positioning information, accommodation information, latitude and longitude, etc.), and other information (a marriage history, beliefs, an
10 undisclosed criminal record, etc.).

[0046] For homomorphic encryption, an operation in a ciphertext domain obtained after decryption is equivalent to a corresponding operation in a plaintext domain. In other words, corresponding computation, for example, an addition operation and a multiplication operation, can still be performed on encrypted data. Therefore, a value obtained after a ciphertext computing
15 result is decrypted is equivalent to a computation result of corresponding plaintext data in plaintext. Generally, homomorphic encryption can be represented by using the following formula:

$$Enc(f(m_1, m_2)) = f(Enc(m_1), Enc(m_2))$$

where m_1 and m_2 represent plaintext data, $Enc(m_1)$ and $Enc(m_2)$ represent ciphertext data, and f represents an operation.

20 **[0047]** The formula indicates that performing an operation on the plaintext m_1 and m_2 after the plaintext is encrypted is equivalent to encrypting the plaintext m_1 and m_2 after an operation is performed on the plaintext. The above formula indicates a basic property of homomorphic encryption, that is, homomorphic encryption is homomorphic to an operation, and a computation result in the ciphertext domain obtained after decryption is equal to the computation result in the
25 plaintext domain.

[0048] The present application also proposes, based on a basic theory of homomorphic encryption, a method for unbalanced PSI. A complex computation process of PSI is converted into a simple matrix transformation operation, which not only ensures ciphertext computing efficiency and retrieval efficiency, but also can implement many advantages such as batch processing and
30 compatibility.

[0049] FIG. 1 is an example block diagram illustrating a procedure of a method for unbalanced PSI according to a first implementation of the present application.

[0050] Referring to FIG. 1, in this implementation, a first party holds a first data set

$Y = \{y_{o1}, y_{o2}, \dots, y_{on}\}$, a second party holds a second data set

5 $X = \{x_{o1}, \dots, x_{om}\}$, and the method for unbalanced PSI is implemented by the first

party holding the first data set. In the methods in this implementation, all ciphertext operations relate to only multiplication and addition, and therefore can be compatible with any somewhat homomorphic encryption solution.

[0051] As shown in FIG. 1, the method for unbalanced PSI includes following steps.

10 [0052] Step S101: Perform data preprocessing on private data in the first data set to obtain a first mapped data set.

[0053] In some implementations, the private data of the first party is a user identifier of a to-be-queried user, and private data of the second party is a group of user identifiers of a target category.

15 [0054] In some implementations, to improve security of an anonymous query, the private data in the first data set is preprocessed by using OPRF to covert the first data set into the first mapped data set. Correspondingly, the private data in the first data set is converted into pseudorandom numbers in the first mapped data set. A subsequent data processing process is specific to the pseudorandom numbers, and has no obvious relationship with the original private data in terms of
20 form.

[0055] Step S103: Obtain a function in a polynomial form through fitting based on the first mapped data set. For example, n pieces of data stored in the first party are fitted to obtain a function $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ in a polynomial form with a highest degree of n .

[0056] In some example implementations, the polynomial with the highest degree of n can be
25 obtained through fitting by using a least squares method/procedure.

[0057] Step S105: Combine coefficients of all terms in the function into a coefficient vector L .

$$L = \begin{bmatrix} a_n \\ a_{n-1} \\ \dots \\ a_1 \\ a_0 \end{bmatrix}$$

[0058] Step S107: Receive a public key PK from the second party, and perform homomorphic encryption on the coefficient vector L by using the public key PK to obtain an encrypted coefficient vector $L_c = Enc(L, PK, e) = PK \cdot (vL) + e$.

5 [0059] In some example implementations, the second party invokes a key generator $KeyGen(\lambda)$ to generate a pair of a private key SK and a public key PK , and performs homomorphic encryption $Enc(L, PK, e)$ on the coefficient vector L based on the public key PK to obtain an encrypted coefficient vector L_c .

10 [0060] The above steps are based on that " $SK \cdot c = vx + e$ " is satisfied between ciphertext and plaintext that undergo homomorphic encryption based on a vector, where e represents an error vector, v represents a large integer, and c and x respectively represent the corresponding ciphertext and plaintext.

[0061] Step S109: Receive a ciphertext input vector X_{ic} from the second party, and obtain a ciphertext result $y_{ic} = X_{ic} \cdot L_c$ through computation with reference to the encrypted coefficient vector L_c .

[0062] In some example implementations, the private key generated by invoking the key generator $KeyGen(\lambda)$ can be represented as $SK = [I, T]P_s$, and the generated public key can be represented as $PK = P_m \begin{bmatrix} I - TA \\ A \end{bmatrix}$, where P_s and P_m are a pair of invertible matrices, $P_s \cdot P_m = I$, I represents an identity matrix, both A and T represent random matrices, and λ represents a security parameter.

20 [0063] In some example implementations, the ciphertext input vector is obtained based on following steps: generating random matrices T' and A' ; generating an identity matrix I' ;

generating a pair of invertible matrices P'_s and P'_m ; and obtaining the ciphertext input vector X_{ic} through computation based on the random matrices T' and A' , the identity matrix I' , the invertible matrix P'_m , and the private key SK of the second party.

[0064] Step S111: Send the ciphertext result to the second party, so that the second party obtains a result of unbalanced PSI.

[0065] FIG. 2 is an example block diagram illustrating a procedure of a method for unbalanced PSI according to a second implementation of the present application.

[0066] Referring to FIG. 2, in this implementation, a first party holds a first data set

$Y = \{y_{o1}, y_{o2}, \dots, y_{on}\}$, a second party holds a second data set $X = \{x_{o1}, \dots, x_{om}\}$, and the method for unbalanced PSI is implemented by the second party holding the second data set.

[0067] As shown in FIG. 2, the method for unbalanced PSI includes following steps.

[0068] Step S201: Perform data preprocessing on private data in the second data set to obtain a second mapped data set.

[0069] In some implementations, private data of the first party is a user identifier of a to-be-queried user, and the private data of the second party is a group of user identifiers of a target category.

[0070] In some implementations, to improve security of an anonymous query, the private data in the second data set is preprocessed by using OPRF to convert the second data set into the second mapped data set. Correspondingly, the private data in the second data set is converted into pseudorandom numbers in the second mapped data set. A subsequent data processing process is specific to the pseudorandom numbers, and has no obvious relationship with the original private data in terms of form.

[0071] Step S203: Generate a pair of a private key and a public key, and send the public key to the first party.

[0072] In some example implementations, the second party invokes a key generator $KeyGen(\lambda)$ to generate a pair of a private key SK and a public key PK . The private key generated by invoking the key generator $KeyGen(\lambda)$ can be represented as

$SK = [I, T]P_s$, and the generated public key can be represented as $PK = P_m \begin{bmatrix} I - TA \\ A \end{bmatrix}$, where P_s and P_m are a pair of invertible matrices, $P_s \cdot P_m = I$, I represents an identity matrix, both A and T represent random matrices, and λ represents a security parameter.

5 [0073] Step S205: Encrypt data in the second mapped data set based on the private key to generate a ciphertext input vector and a decryption key.

[0074] In some example implementations, the ciphertext input vector is obtained based on following steps: generating random matrices T' and A' ; generating an identity matrix I' ; generating a pair of invertible matrices P'_s and P'_m ; and obtaining the ciphertext input vector
10 X_{ic} through computation based on the random matrices T' and A' , the identity matrix I' , the invertible matrix P'_m , and the private key SK of the second party.

[0075] In some example implementation, the decryption key is generated based on following steps: generating random matrices T' and A' ; generating an identity matrix I' ; generating a pair of invertible matrices P'_s and P'_m ; and generating the decryption key
15 $SK' = [I', T'] \cdot P'_s$ through computation based on the random matrix T' , the identity matrix I' , and the invertible matrix P'_s .

[0076] Step S207: Send the ciphertext input vector to the first party.

[0077] Step S209: Receive a ciphertext result based on the ciphertext input vector from the first party, and decrypt the ciphertext result by using the decryption key to obtain a result of
20 unbalanced PSI.

[0078] In some example implementations, the ciphertext result y_{ic} is obtained through computation based on the ciphertext input vector X_{ic} and the encrypted coefficient vector L_c .

$$y_{ic} = X_{ic} \cdot L_c$$

25 [0079] It can be appreciated from the above mathematical expression of the ciphertext result

that, the ciphertext input vector X_{ic} is an encrypted data matrix, and the encrypted coefficient vector L_c is also an encrypted data matrix. When the ciphertext result is computed, no polynomial with a degree up to n appears. Therefore, in the present application, computation of a polynomial with any number of degrees can be converted into one linear transformation operation, which not only ensures ciphertext computing efficiency and security, but also improves retrieval query efficiency.

[0080] For the unbalanced PSI, the first party performs data preprocessing based on OPRF, and establishes an interpolation polynomial $f(x) = (x - y_1)(x - y_2) \cdots (x - y_n)$ based on the preprocessed first mapped data set. Data x_i of the second party used as query data is used as an input of the interpolation polynomial to compute $f(x_i)$. If $f(x_i)$ is equal to 0, certain data y equal to x_i necessarily exists in $\{y_1, y_2, \cdots, y_n\}$, that is, y and x_i are intersection data in the first data set held by the first party and the second data set held by the second party.

[0081] In some example implementations, when receiving the ciphertext result y_{ic} based on the ciphertext input vector X_{ic} from the first party, the second party obtains the result of the unbalanced PSI through computation by using the decryption key SK' .

$$y = \frac{SK' \cdot y_{ic}}{v}$$

[0082] If $y = 0$, the pseudorandom number x_i in the second mapped data set is one piece of intersection data, and correspondingly, the private data x_{oi} that is in the second data set and that is mutually mapped to the pseudorandom number x_i is the intersection data.

[0083] If $y \neq 0$, the pseudorandom number x_i in the second mapped data set is not one piece of intersection data, and correspondingly, the private data x_{oi} that is in the second data set and that is mutually mapped to the pseudorandom number x_i is not the intersection data.

[0084] In some implementation, the method in the present application can also be used to

implement batch query processing of a user, to be specific, a same user determines, at one time, whether a plurality of pieces of data $(x_1, x_2, \dots, x_m)$ are intersection data. In this case, a result of determining whether the plurality of pieces of data are the intersection data can be obtained based on following steps:

- 5 computing a vector matrix X in plaintext based on degree information $[n, n-1, \dots, 1, 0]$ of the function $f(x)$ sent by the first party, where

$$X = \begin{bmatrix} x_1^n, x_1^{n-1}, \dots, x_1^1, 1 \\ x_2^n, x_2^{n-1}, \dots, x_2^1, 1 \\ \dots \\ x_m^n, x_m^{n-1}, \dots, x_m^1, 1 \end{bmatrix};$$

randomly generating an invertible matrix pair $P'_s \cdot P'_m = I$;

randomly generating matrices T' and A' ;

- 10 generating the decryption key $SK' = [I', T'] \cdot P'_s$;

generating a ciphertext query vector $X_c = P'_m (X \cdot SK_u^{-T'} A')$; and

sending the ciphertext query vector X_c to the first party for batch query processing.

[0085] FIG. 3 is an example schematic diagram illustrating a system architecture of an apparatus for unbalanced PSI according to a third implementation of the present application.

- 15 [0086] Referring to FIG. 3, in this implementation, a first party holds a first data set $Y = \{y_{o1}, y_{o2}, \dots, y_{on}\}$, and a second party holds a second data set $X = \{x_{o1}, \dots, x_{om}\}$. The apparatus for unbalanced PSI is disposed in the first party.

[0087] As shown in FIG. 3, the apparatus includes a preprocessing unit 30, a fitting unit 32, an encryption unit 34, a computation unit 36, and a sending unit 38.

- 20 [0088] The preprocessing unit 30 is configured to perform data preprocessing on private data in the first data set to obtain a first mapped data set.

[0089] In some implementations, the private data of the first party is a user identifier of a to-be-queried user, and private data of the second party is a group of user identifiers of a target category.

[0090] In some implementations, to improve security of an anonymous query, the private data in the first data set is preprocessed by using OPRF to covert the first data set into the first mapped data set. Correspondingly, the private data in the first data set is converted into pseudorandom numbers in the first mapped data set. A subsequent data processing process is specific to the pseudorandom numbers, and has no obvious relationship with the original private data in terms of form.

[0091] The fitting unit 32 is configured to: obtain a function in a polynomial form through fitting based on the first mapped data set, and combine coefficients of all terms in the function into a coefficient vector. For example, the fitting unit 32 fits n pieces of data stored in the first party to obtain a function $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ in a polynomial form with a highest degree of n .

[0092] In some example implementations, the polynomial with the highest degree of n can be obtained through fitting by using a least squares method/procedure.

[0093] The encryption unit 34 is configured to: receive a public key PK from the second party, and perform homomorphic encryption on the coefficient vector L by using the public key PK to obtain an encrypted coefficient vector

$$L_c = \text{Enc}(L, PK, e) = PK \cdot (vL) + e.$$

[0094] In some example implementations, the second party invokes a key generator $\text{KeyGen}(\lambda)$ to generate a pair of a private key SK and a public key PK , and performs homomorphic encryption $\text{Enc}(L, PK, e)$ on the coefficient vector L based on the public key PK to obtain an encrypted coefficient vector L_c .

[0095] The above steps are based on that " $SK \ c = vx + e$ " is satisfied between ciphertext and plaintext that undergo homomorphic encryption based on a vector, where e represents an error vector, v represents a large integer, and c and x respectively represent the corresponding ciphertext and plaintext.

[0096] The computation unit 36 is configured to: receive a ciphertext input vector X_{ic} from the second party, and obtain a ciphertext result $y_{ic} = X_{ic} \cdot L_c$ through computation with reference to the encrypted coefficient vector L_c .

[0097] In some example implementations, the private key generated by invoking the key

generator $KeyGen(\lambda)$ can be represented as $SK = [I, T]P_s$, and the generated public key can be represented as $PK = P_m \begin{bmatrix} I - TA \\ A \end{bmatrix}$, where P_s and P_m are a pair of invertible matrices, $P_s \cdot P_m = I$, I represents an identity matrix, both A and T represent random matrices, and λ represents a security parameter.

5 [0098] In some example implementation, the ciphertext input vector is obtained based on following steps: generating random matrices T' and A' ; generating an identity matrix I' ; generating a pair of invertible matrices P'_s and P'_m ; and obtaining the ciphertext input vector X_{ic} through computation based on the random matrices T' and A' , the identity matrix I' , the invertible matrix P'_m , and the private key SK of the second party.

10 [0099] The sending unit 38 is configured to send the ciphertext result to the second party, so that the second party obtains a result of unbalanced PSI.

[0100] FIG. 4 is an example schematic diagram illustrating a system architecture of an apparatus for unbalanced PSI according to a fourth implementation of the present application.

[0101] Referring to FIG. 4, in this implementation, a first party holds a first data set
15 $Y = \{y_{o1}, y_{o2}, \dots, y_{on}\}$, and a second party holds a second data set $X = \{x_{o1}, \dots, x_{om}\}$. The apparatus for unbalanced PSI is disposed in the second party.

[0102] As shown in FIG. 4, the apparatus includes a preprocessing unit 41, a key generation unit 43, an encryption unit 45, a sending unit 47, and a decryption unit 49.

20 [0103] The preprocessing unit 41 is configured to perform data preprocessing on private data in the second data set to obtain a second mapped data set.

[0104] In some implementations, private data of the first party is a user identifier of a to-be-queried user, and the private data of the second party is a group of user identifiers of a target category.

25 [0105] In some implementations, to improve security of an anonymous query, the private data in the second data set is preprocessed by using OPRF to convert the second data set into the second mapped data set. Correspondingly, the private data in the second data set is converted into

pseudorandom numbers in the second mapped data set. A subsequent data processing process is specific to the pseudorandom numbers, and has no obvious relationship with the original private data in terms of form.

[0106] The key generation unit 43 is configured to: generate a pair of a private key and a public key based on a homomorphic encryption algorithm, and send the public key to the first party.

[0107] In some example implementations, the second party invokes a key generator $KeyGen(\lambda)$ to generate a pair of a private key SK and a public key PK . The private key

generated by invoking the key generator $KeyGen(\lambda)$ can be represented as

$$SK = [I, T]P_s, \text{ and the generated public key can be represented as } PK = P_m \begin{bmatrix} I - TA \\ A \end{bmatrix}, \text{ where } P_s \text{ and } P_m \text{ are a pair of invertible matrices, } P_s \cdot P_m = I,$$

I represents an identity matrix, both A and T represent random matrices, and λ represents a security parameter.

[0108] The encryption unit 45 is configured to encrypt data in the second mapped data set based on the private key to generate a ciphertext input vector and a decryption key.

[0109] In some example implementations, the ciphertext input vector is obtained based on following steps: generating random matrices T' and A' ; generating an identity matrix I' ; generating a pair of invertible matrices P'_s and P'_m ; and obtaining the ciphertext input vector X_{ic} through computation based on the random matrices T' and A' , the identity matrix I' , the invertible matrix P'_m , and the private key SK of the second party.

[0110] In some example implementations, the decryption key is generated based on following steps: generating random matrices T' and A' ; generating an identity matrix I' ; generating a pair of invertible matrices P'_s and P'_m ; and generating the decryption key $SK' = [I', T'] \cdot P'_s$ through computation based on the random matrix T' , the identity matrix I' , and the invertible matrix P'_s .

[0111] The sending unit 47 is configured to send the ciphertext input vector to the first party.

[0112] The decryption unit 49 is configured to: receive a ciphertext result based on the

ciphertext input vector from the first party, and decrypt the ciphertext result by using the decryption key to obtain a result of unbalanced PSI.

[0113] In some example implementations, the ciphertext result y_{ic} is obtained through computation based on the ciphertext input vector X_{ic} and the encrypted coefficient vector L_c .

$$y_{ic} = X_{ic} \cdot L_c$$

[0114] It can be appreciated from the above mathematical expression of the ciphertext result that, the ciphertext input vector X_{ic} is an encrypted data matrix, and the encrypted coefficient vector L_c is also an encrypted data matrix. When the ciphertext result is computed, no polynomial with a degree up to n appears. Therefore, in the present application, computation of a polynomial with any number of degrees can be converted into one linear transformation operation, which not only ensures ciphertext computing efficiency and security, but also improves retrieval query efficiency.

[0115] According to the principle of the unbalanced PSI, the first party performs data preprocessing based on OPRF, and establishes an interpolation polynomial $f(x) = (x - y_1)(x - y_2) \cdots (x - y_n)$ based on the preprocessed first mapped data set. Data x_i of the second party used as query data is used as an input of the interpolation polynomial to compute $f(x_i)$. If $f(x_i)$ is equal to 0, certain data y equal to x_i necessarily exists in $\{y_1, y_2, \cdots, y_n\}$, that is, y and x_i are intersection data in the first data set held by the first party and the second data set held by the second party.

[0116] In some example implementations, when receiving the ciphertext result y_{ic} based on the ciphertext input vector X_{ic} from the first party, the second party obtains the result of the unbalanced PSI through computation by using the decryption key SK' .

$$y = \frac{SK' \cdot y_{ic}}{v}$$

[0117] If $y = 0$, the pseudorandom number x_i in the second mapped data set is one

piece of intersection data, and correspondingly, the private data x_{oi} that is in the second data set and that is mutually mapped to the pseudorandom number x_i is the intersection data.

[0118] If $y \neq 0$, the pseudorandom number x_i in the second mapped data set is not one piece of intersection data, and correspondingly, the private data x_{oi} that is in the second data set and that is mutually mapped to the pseudorandom number x_i is not the intersection data.

[0119] To verify technical effects of the method and the apparatus for unbalanced PSI in the present application, a decryption process is expanded herein as:

$$\begin{aligned}
 &= SK' \cdot y_{ic} \\
 &= SK' \cdot X_{ic} \cdot L_c \\
 &= [I', T'] \cdot P'_s \cdot P'_m \cdot \begin{bmatrix} X_i \cdot SK - T' A' \\ A' \end{bmatrix} \cdot PK \cdot (vL) + e' \\
 &= X_i \cdot SK \cdot PK \cdot (vL) + e' \\
 &= v(X_i \cdot L) + e' \\
 &= v([x_i^n, x_i^{n-1}, x_i \cdots, x_i, 1] \cdot \begin{bmatrix} a_n \\ a_{n-1} \\ \cdots \\ a_1 \\ a_0 \end{bmatrix}) + e' \\
 &= v(a_n x_i^n + a_{n-1} x_i^{n-1} + \cdots a_1 x_i + a_0) + e' \\
 &\quad \frac{e'}{v} \approx 0
 \end{aligned}$$

10 where e' in the above formula represents small noise, and $\frac{e'}{v} \approx 0$.

[0120] The above result is divided by a big integer v to obtain $a_n x_i^n + a_{n-1} x_i^{n-1} + \cdots a_1 x_i + a_0$, that is, a final decryption result y .

Therefore, it can be appreciated that the decryption result is equal to a computation result of a function in a polynomial form in plaintext, and is also equal to a query result.

15 [0121] The same method is used to verify technical effects of determining, by a same user at one time, whether a plurality of pieces of data are intersection data. A final decryption form is:

$$v\left(\begin{bmatrix} x_1^n, x_1^{n-1}, \dots, x_1^1, 1 \\ x_2^n, x_2^{n-1}, \dots, x_2^1, 1 \\ \dots \\ x_m^n, x_m^{n-1}, \dots, x_m^1, 1 \end{bmatrix} \cdot \begin{bmatrix} a^n \\ a^{n-1} \\ \dots \\ a \\ a_0 \end{bmatrix}\right)$$

[0122] A result is equal to $\begin{bmatrix} y_1 = f(x_1) \\ y_2 = f(x_2) \\ \dots \\ y_m = f(x_m) \end{bmatrix}$.

[0123] Therefore, it can be appreciated that when batch query processing is performed, the decryption result is also equal to a computation result of a function in a polynomial form in plaintext, and is also equal to a query result. This verification also indicates that batch query processing is correctly completed by using the method and the apparatus in the present application.

[0124] FIG. 5 is an example diagram illustrating a structure of a computer-readable storage medium corresponding to a method for unbalanced PSI according to another implementation of the present application.

[0125] Referring to FIG. 5, this implementation further provides a computer-readable storage medium. The computer-readable storage medium includes one or more computer programs 501. These computer programs 501 store instructions. When the instructions are run on a processor, the above method for unbalanced PSI is performed. For example, the computer instructions can include computer program code, and the computer program code can be in a source code form, an object code form, an executable file, or some intermediate forms.

[0126] The processor can include one or more processing units. For example, the processor can include one or more of an application processor (AP), a modem processor, a graphics processing unit (GPU), an image signal processor (ISP), a controller, a memory, a video codec, a digital signal processor (DSP), a baseband processor, and a neural-network processing unit (NPU).

[0127] Different processing units can be independent components, or can be integrated into one or more processors.

[0127] The computer-readable medium shown in the present invention can be a computer-readable signal medium, a computer-readable storage medium, or any combination of the two. For example, the computer-readable storage medium can be, but is not limited to, an electrical,

magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or component, or any combination thereof. More specific examples of the computer-readable storage medium can include but are not limited to: an electrical connection having one or more wires, a portable computer disk, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or a flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage component, a magnetic storage component, or any appropriate combination thereof. In the present invention, the computer-readable storage medium can be any tangible medium that includes or stores a program, and the program can be used by or in combination with an instruction execution system, apparatus, or component. In the present invention, the computer-readable signal medium can include a data signal propagated in a baseband or as a part of a carrier, and includes computer-readable program code. The propagated data signal can be in a plurality of forms, including but not limited to an electromagnetic signal, an optical signal, or any appropriate combination thereof. The computer-readable signal medium can alternatively be any computer-readable medium other than the computer-readable storage medium, and the computer-readable medium can send, propagate, or transmit a program used by or in combination with an instruction execution system, apparatus, or component. The program code included in the computer-readable medium can be transmitted by using any appropriate medium, including but not limited to wireless, a cable, an optical cable, RF, or any appropriate combination thereof.

[0128] It can be appreciated from the above that, in the implementations of the present specification, the first party holding the first data set can obtain, from the second party holding the second data set, only the ciphertext input vector obtained through homomorphic encryption, and the first party cannot infer any data included in the second data set by using the ciphertext input vector. The first party sends the ciphertext result to the second party, and the ciphertext result is also obtained by the first party through linear transformation computation by using two encryption matrices. The second party decrypts the ciphertext result by using the decryption key, and can only determine, based on the decrypted ciphertext result, whether the private data of the first party belongs to the second data set, and cannot inversely deduce and determine a specific location of to-be-queried data in the data set, so that a security requirement of an anonymous query can be satisfied. Compared with the existing technologies, in the method for unbalanced PSI in this solution, a complex computation process of PSI is converted into a simple matrix transformation

operation, which not only ensures ciphertext computing efficiency and retrieval efficiency, but also can implement many advantages such as batch processing and compatibility.

[0129] Example implementations of the present application are described above with reference to the accompanying drawings. However, a person of ordinary skill in the art can understand that, 5 various changes and replacements can be made to the example implementations of the present application without departing from the spirit and scope of the present application. These changes and replacements all fall within the scope defined by the claims of the present application.

CLAIMS

What is claimed is:

1. A method for unbalanced private set intersection (PSI), a first party having a first data set, a second party having a second data set, and the method being performed by the first party and
5 comprising:
 - performing data preprocessing on private data in the first data set to obtain a first mapped data set;
 - obtaining a function in a polynomial form through fitting based on the first mapped data set;
 - combining coefficients of all terms in the function into a coefficient vector;
 - 10 receiving a public key from the second party, and performing homomorphic encryption on the coefficient vector by using the public key to obtain an encrypted coefficient vector;
 - receiving a ciphertext input vector from the second party, and obtaining a ciphertext result through computation with reference to the encrypted coefficient vector; and
 - sending the ciphertext result to the second party, so that the second party obtains a result of
15 unbalanced PSI.
2. The method according to claim 1, wherein the performing data preprocessing on the first data set comprises:
 - performing data preprocessing by using an oblivious pseudorandom function (OPRF),
20 wherein all data in the first mapped data set are pseudorandom numbers.
3. The method according to claim 1, wherein the private data of the first party is a user identifier of a to-be-queried user, and private data of the second party is a group of user identifiers of a target category.
25
4. The method according to claim 1, wherein the ciphertext input vector is obtained based on following steps:
 - generating a random matrix;
 - generating an identity matrix;
 - 30 generating an invertible matrix pair; and

obtaining the ciphertext input vector through computation based on the random matrix, the identity matrix, the invertible matrix pair, and a private key of the second party.

5 5. The method according to claim 1, wherein the function is obtained through fitting by using a least squares procedure.

6. A method for unbalanced private set intersection (PSI), a first party holding a first data set, a second party holding a second data set, and the method being performed by the second party and comprising:

10 performing data preprocessing on private data in the second data set to obtain a second mapped data set;

 generating a pair of a private key and a public key, and sending the public key to the first party;

 encrypting data in the second mapped data set based on the private key to generate a
15 ciphertext input vector and a decryption key;

 sending the ciphertext input vector to the first party; and

 receiving a ciphertext result based on the ciphertext input vector from the first party, and decrypting the ciphertext result by using the decryption key to obtain a result of unbalanced PSI.

20 7. The method according to claim 6, wherein the preprocessing is performed by using an oblivious pseudorandom function (OPRF), and all data in the second mapped data set are pseudorandom numbers.

 8. The method according to claim 6, wherein the ciphertext input vector is generated based
25 on following steps:

 generating a random matrix;

 generating an identity matrix;

 generating an invertible matrix pair; and

 obtaining the ciphertext input vector through computation based on the private key, the
30 random matrix, the identity matrix, and the invertible matrix pair.

9. The method according to claim 6, wherein the decryption key is generated based on following steps:

generating a random matrix;

generating an identity matrix;

5 generating an invertible matrix pair; and

generating the decryption key through computation based on the random matrix, the identity matrix, and the invertible matrix pair.

10 10. The method according to claim 6, wherein if a result of the unbalanced PSI obtained through the decrypting is 0, private data of the first party is intersection data; and if the result of the unbalanced PSI obtained through the decrypting is not 0, the private data of the first party is not intersection data.

15 11. The method according to claim 6, wherein private data of the first party is a user identifier of a to-be-queried user, and the private data of the second party is a group of user identifiers of a target category.

20 12. An apparatus for unbalanced private set intersection (PSI), a first party holding a first data set, a second party holding a second data set, and the apparatus being disposed in the first party and comprising:

a preprocessing unit, configured to perform data preprocessing on private data in the first data set to obtain a first mapped data set;

25 a fitting unit, configured to: obtain a function in a polynomial form through fitting based on the first mapped data set, and combine coefficients of all terms in the function into a coefficient vector;

an encryption unit, configured to: receive a public key from the second party, and perform homomorphic encryption on the coefficient vector by using the public key to obtain an encrypted coefficient vector;

30 a computation unit, configured to: receive a ciphertext input vector from the second party, and obtain a ciphertext result through computation with reference to the encrypted coefficient vector; and

a sending unit, configured to send the ciphertext result to the second party, so that the second party obtains a result of unbalanced PSI.

13. The apparatus according to claim 12, wherein the preprocessing unit performs data preprocessing by using an oblivious pseudorandom function (OPRF), and all data in the first mapped data set are pseudorandom numbers.

14. The apparatus according to claim 12, wherein the function is obtained through fitting by using a least squares method/procedure.

10

15. The apparatus according to claim 12, wherein the private data of the first party is a user identifier of a to-be-queried user, and private data of the second party is a group of user identifiers of a target category.

16. An apparatus for unbalanced private set intersection (PSI), a first party holding a first data set, a second party holding a second data set, and the apparatus being disposed in the second party and comprising:

a preprocessing unit, configured to perform data preprocessing on private data in the second data set to obtain a second mapped data set;

20 a key generation unit, configured to: generate a pair of a private key and a public key based on a homomorphic encryption algorithm, and send the public key to the first party;

an encryption unit, configured to encrypt data in the second mapped data set based on the private key to generate a ciphertext input vector and a decryption key;

a sending unit, configured to send the ciphertext input vector to the first party; and

25 a decryption unit, configured to: receive a ciphertext result based on the ciphertext input vector from the first party, and decrypt the ciphertext result by using the decryption key to obtain a result of unbalanced PSI.

17. The apparatus according to claim 16, wherein the preprocessing unit performs data preprocessing by using an oblivious pseudorandom function (OPRF), and all data in the second mapped data set are pseudorandom numbers.

18. The apparatus according to claim 16, wherein the encryption unit obtains the ciphertext input vector based on following steps:

generating a random matrix;

5 generating an identity matrix;

generating an invertible matrix pair; and

obtaining the ciphertext input vector through computation based on the private key, the random matrix, the identity matrix, and the invertible matrix pair.

10 19. The apparatus according to claim 16, wherein the encryption unit generates the decryption key based on following steps:

generating a random matrix;

generating an identity matrix;

generating an invertible matrix pair; and

15 generating the decryption key through computation based on the random matrix, the identity matrix, and the invertible matrix pair.

20. The apparatus according to claim 16, wherein if the result of the unbalanced PSI obtained through the decrypting is 0, it is determined that private data of the first party is intersection data;

20 and if the result of the unbalanced PSI obtained through the decrypting is not 0, it is determined that the private data of the first party is not the intersection data.