

Data Transmission Method, Apparatus and System

Technical Field

The present application relates to the technical field of network communications, particularly to a data transmission method, apparatus, and system.

5 **Background Art**

Nowadays, people are paying increasing attention to data security, particularly data security during transmission. Taking offline payments as an example, a server device may send a strategy for generation of payment codes to a client device, and the client device stores the strategy. When a user needs to use a payment code, the client device may use the strategy to
10 generate a payment code. A merchant scans the payment code by a scanning device. The scanning device transmits information obtained from scanning to the server device for verification. After the information passes the verification, money is deducted. It can be seen that, in a process when a server device transmits the strategy to a client device, security of the channel between the client device and the server device needs to be ensured. If the strategy
15 issued by the server device is intercepted by a third-party hacker, serious losses will be incurred by the user of the client device.

In a related art, an identical key may be preset on all client devices and server devices. A server device may use the key to encrypt information to be transmitted and transmit a ciphertext to a client device. The client device uses the key to decrypt the ciphertext. However, as all the
20 client devices and server devices share the same key, if the key of a client device or a server device leaks, all the client devices and server devices will be at a security risk.

In another related art, a client device may generate a pair of asymmetrical keys, save a private key, and upload a public key to a server device. The server device uses the public key to encrypt information that needs to be transmitted and transmits a ciphertext to the client
25 device. The client device uses the private key to decrypt the ciphertext. As the asymmetrical key algorithm uses a different random number during each calculation, a different asymmetrical key pair is generated during each calculation. Therefore, asymmetrical key pairs generated by different clients are different, too, avoiding the problem of a security risk for all client devices and server devices resulting from leakage of the key of a client device or a server
30 device. Meanwhile, as the ciphertext can be decrypted only by the private key to which the public key pair corresponds, even if the public key is intercepted during transmission of the

public key, the ciphertext still cannot be decrypted through the public key, thereby ensuring information security. However, an asymmetrical key needs to use a complex encryption algorithm for encryption and a complex decryption algorithm for decryption, so encryption and decryption take a long time.

5 Summary

The present application provides a data transmission method, apparatus, and system to solve the problems of information security and prolonged encryption and decryption in current technologies.

According to a first aspect of an embodiment of the present application, a data transmission method is provided, the method being applicable in a client and comprises:

generating an asymmetrical key pair comprising a first public key and a first private key, and sending a data request carrying the first public key to a server;

receiving a ciphertext and a second public key sent by the server, wherein the second public key is a public key in an asymmetrical key pair obtained by the server, the asymmetrical key pair obtained by the server further comprises a second private key, and the ciphertext is information obtained by encrypting a seed parameter for generating an offline payment code using a shared key; the shared key is a key generated based on the second private key and the first public key using a preset key-agreement algorithm;

generating a shared key based on the first private key and the second public key using the key-agreement algorithm, and using the shared key to decrypt the ciphertext to obtain the seed parameter;

Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key based on the first private key and the second public key generated using the key-agreement algorithm.

According to a second aspect of an embodiment of the present application, a data transmission method is provided, the method being applicable in a server and comprises:

receiving a data request carrying a first public key and sent by a client, wherein the data request is for requesting the server to return a seed parameter for generating an offline payment code, the first public key is a public key in an asymmetrical key pair generated by the client, and the asymmetrical key pair generated by the client further comprises a first private key;

obtaining an asymmetrical key pair comprising a second public key and a second private key, and generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

using the shared key to encrypt a seed parameter to which the data request corresponds,
 5 and sending a ciphertext obtained from encryption and the second public key to the client so that the client generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the seed parameter.

Here, the shared key generated based on the second private key and the first public key
 10 using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

According to a third aspect of an embodiment of the present application, a data transmission method is provided and comprises:

a data requester terminal generates an asymmetrical key pair comprising a first public
 15 key and a first private key and sends a data request carrying the first public key to a data provider terminal;

the data provider terminal obtains an asymmetrical key pair comprising a second public key and a second private key and generates a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

20 the data provider terminal uses the shared key to encrypt target data to which the data request corresponds and sends a ciphertext obtained from encryption and the second public key to the data requester terminal;

the data requester terminal generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the
 25 ciphertext to obtain the target data;

Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

According to a fourth aspect of an embodiment of the present application, a data
 30 transmission apparatus is provided and comprises:

a key generation module for generating an asymmetrical key pair comprising a first public key and a first private key;

a request sending module for sending a data request carrying the first public key to a server;

5 an information receiving module for receiving a ciphertext and a second public key sent by the server, wherein the second public key is a public key in an asymmetrical key pair obtained by the server, the asymmetrical key pair obtained by the server further comprises a second private key, and the ciphertext is information obtained by encrypting a seed parameter for generating an offline payment code using a shared key; the shared key is a key generated
10 based on the second private key and the first public key using a preset key-agreement algorithm;

a shared key generation module for generating a shared key based on the first private key and the second public key using the key-agreement algorithm; and

an information decryption module for using the shared key to decrypt the ciphertext to obtain the seed parameter.

15 Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

According to a fifth aspect of an embodiment of the present application, a data transmission apparatus is provided and comprises:

20 a request receiving module for receiving a data request carrying a first public key and sent by a client, wherein the data request is for requesting a server to return a seed parameter for generating an offline payment code, the first public key is a public key in an asymmetrical key pair generated by the client, and the asymmetrical key pair generated by the client further comprises a first private key;

25 a key obtaining module for obtaining an asymmetrical key pair comprising a second public key and a second private key;

a shared key generation module for generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

30 an information encryption module for using the shared key to encrypt a seed parameter to which the data request corresponds;

an information sending module for sending a ciphertext obtained from encryption and the second public key to the client so that the client generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the seed parameter;

5 Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

According to a sixth aspect of an embodiment of the present application, a data transmission system is provided and comprises a data requester device and a data provider
10 device;

The data requester device generates an asymmetrical key pair comprising a first public key and a first private key and sends a data request carrying the first public key to a data provider device;

The data provider device obtains an asymmetrical key pair comprising a second public
15 key and a second private key and generates a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

The data provider device uses the shared key to encrypt target data to which the data request corresponds and sends a ciphertext obtained from encryption and the second public key to the data requester device;

20 The data requester device generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the target data;

Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the
25 first private key and the second public key using the key-agreement algorithm.

According to a seventh aspect of an embodiment of the present application, a data transmission method is provided and comprises:

a data requester terminal generates a first symmetric key and sends a data request carrying the first symmetric key to a data provider terminal;

30 the data provider terminal obtains a second symmetric key and generates a shared key

based on the first symmetric key and the second symmetric key using a preset key-agreement algorithm, the second symmetric key being different from the first symmetric key;

the data provider terminal uses the shared key to encrypt target data to which the data request corresponds and sends a ciphertext obtained from encryption and the second symmetric key to the data requester terminal;

the data requester terminal generates a shared key based on the first symmetric key and the second symmetric key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the target data.

According to an eighth aspect of an embodiment of the present application, a data transmission system is provided and comprises a data requester device and a data provider device;

the data requester device generates a first symmetric key and sends a data request carrying the first symmetric key to a data provider device;

the data provider device obtains a second symmetric key and generates a shared key based on the first symmetric key and the second symmetric key using a preset key-agreement algorithm, the second symmetric key being different from the first symmetric key;

the data provider device uses the shared key to encrypt target data to which the data request corresponds and sends a ciphertext obtained from encryption and the second symmetric key to the data requester device;

the data requester device generates a shared key based on the first symmetric key and the second symmetric key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the target data.

During application of the data transmission method, apparatus, and system provided in embodiments of the present application, an asymmetrical key pair comprising a first public key and a first private key may be generated through a data requester terminal, a data request carrying the first public key is sent to a data provider terminal, an asymmetrical key pair comprising a second public key and a second private key is obtained through a data provider terminal, a shared key is generated based on the second private key and the first public key using a preset key-agreement algorithm, then the shared key is used to encrypt target data to which the data request corresponds, and lastly a ciphertext obtained from encryption and the second public key are transmitted to the data requester terminal, and the data requester terminal

generates a shared key based on the first private key and the second public key using the same key-agreement algorithm. As the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm, the data provider terminal may use the shared key for encryption, and the data requester terminal may use the shared key for decryption. As the key for encryption of target data and the key for decryption of target data are identical, symmetric encryption and decryption algorithms may be used to encrypt and decrypt data. As a symmetric encryption algorithm typically conducts encryption by such means as shift cipher, while an asymmetrical encryption algorithm conducts encryption by such methods as finding large prime numbers, it can be seen that the encryption process of a symmetric encryption algorithm is simpler than the encryption process of an asymmetrical encryption algorithm. Therefore, this embodiment may avoid the defect of long encryption and decryption times resulting from complex asymmetrical encryption and decryption algorithms and improve encryption and decryption efficiency. Further, as the complete key is not exposed through the entire transmission process, it is meaningless even if a public key is hijacked by a hacker, thereby ensuring data security throughout the entire transmission process.

During application of the data transmission method, apparatus, and system provided in embodiments of the present application, a first symmetric key may be obtained through a data requester terminal, a data request carrying the first symmetric key is sent to a data provider terminal, a second symmetric key is obtained through a data provider terminal, a shared key is generated based on the first symmetric key and the second symmetric key using a preset key-agreement algorithm, the shared key is used to encrypt target data to which the data request corresponds, and lastly a ciphertext obtained from encryption and the second symmetric key are transmitted to the data requester terminal, and the data requester terminal uses the same key-agreement algorithm to generate a shared key based on the first symmetric key and the second symmetric key. As the data provider terminal and the data requester terminal use an identical key-agreement algorithm, the shared key generated by the data provider terminal and that generated by data requester terminal are identical, and the data requester terminal may decrypt the ciphertext through the generated shared key, thereby obtaining the target data. It can be seen that, as the shared key is different from the first symmetric key and the second symmetric key, even if a hacker has hijacked the symmetric key, the hacker will not know what key-agreement algorithm the present application has used, so the hacker is unable to decrypt

the ciphertext, thereby ensuring data security throughout the entire transmission process. Further, as the key for encryption of target data and the key for decryption of target data are identical, symmetric key encryption and decryption algorithms are used to encrypt and decrypt data to avoid long encryption and decryption times resulting from complex asymmetrical encryption and decryption algorithms, thereby improving encryption and decryption efficiency.

It should be understood that the foregoing general description and subsequent detailed description are exemplary and explanatory only and cannot limit the present application.

Brief Description of the Drawings

The accompanying drawings here are included into and constitute a part of the Description, show embodiments conforming to the present application, and are used together with the Description to explain the principles of the present application.

FIG. 1A is a schematic diagram of an application scenario of data transmission provided by an exemplary embodiment of the present application.

FIG. 1B is a flow chart of an embodiment of a data transmission method provided by the present application.

FIG. 2 is a flow chart of another embodiment of a data transmission method provided by the present application.

FIG. 3 is a flow chart of another embodiment of a data transmission method provided by the present application.

FIG. 4 is a block diagram of an embodiment of a data transmission system provided by the present application.

FIG. 5 is a block diagram of an embodiment of a data transmission apparatus provided by the present application.

FIG. 6 is a block diagram of another embodiment of a data transmission apparatus provided by the present application.

FIG. 7 is a flow chart of another embodiment of a data transmission method provided by the present application.

FIG. 8 is a block diagram of another embodiment of a data transmission system provided

by the present application.

Detailed Description

Here, exemplary embodiments will be described in detail, with examples shown in the accompanying drawings. When the description below involves the accompanying drawings, unless otherwise indicated, the same numeral in different accompanying drawings stands for the same or similar element. The implementation manners described in the following exemplary embodiments do not represent all the implementation manners consistent with the present application. Conversely, they are only examples of the apparatus and method described in detail in the attached claims and consistent with some aspects of the present application.

The terms used in the present application are only for the purpose of describing exemplary embodiments and not intended to limit the present application. The singular forms “one”, “the”, and “this” used in the present application and in the attached claims also are intended to cover plural forms unless their meanings are otherwise clearly indicated in the context. It should also be understood that the term “and/or” used in the text refers to any or all possible combinations containing one or a plurality of the associated listed items.

It should be understood that although the present application may use terms such as first, second, and third to describe various kinds of information, the information should not be limited to these terms. These terms are only intended to differentiate information of the same type. For example, without departing from the scope of the present application, first information may also be referred to as second information, and similarly, second information may also be referred to as first information. Subject to the context, the term “if” used here may be interpreted as “at the time of ...”, “when ...”, or “in response to a determination.”

People are paying increasing attention to data security, particularly to data security during transmission. As shown in FIG. 1A, FIG. 1A is a schematic diagram of an application scenario of data transmission provided by an exemplary embodiment of the present application. In this schematic diagram, data transmission may be conducted between different client devices and server devices. For example, a client device sends a data request to a server device, and the server device returns corresponding target data according to the data request. In the transmission process, hackers might intercept the target data that is being transmitted, thereby causing losses to users.

In order to ensure data security during transmission, an identical key may be preset on

all client devices and server devices. A server device may use the key to encrypt information to be transmitted and transmit a ciphertext to a client device. The client device uses the key to decrypt the ciphertext. However, as all the client devices and server devices share the same key, if the key of a client device or a server device leaks, all the client devices and server devices will be at a security risk. In order to avoid this circumstance, in another related art, a client device may generate a pair of asymmetrical keys, save a private key, and upload a public key to a server device. The server device uses the public key to encrypt information that needs to be transmitted and transmits a ciphertext to the client device. The client device uses the private key to decrypt the ciphertext. As the asymmetrical key algorithm uses a different random number during each calculation, a different asymmetrical key pair is generated during each calculation. Therefore, asymmetrical key pairs generated by different clients are different, too, avoiding the problem of a security risk for all client devices and server devices resulting from leakage of the key of a client device or a server device. Meanwhile, as the ciphertext can be decrypted only by a private key to which a public key pair corresponds, even if the public key is intercepted during transmission of the public key, the ciphertext still cannot be decrypted through the public key, thereby ensuring information security. However, an asymmetrical key needs to use a complex encryption algorithm for encryption and a complex decryption algorithm for decryption, so encryption and decryption take a long time.

In order to avoid the information security problems in current technologies and the problem of encryption and decryption taking a long time, the present application provides a data transmission method, as shown in FIG. 1B. FIG. 1B is a flow chart of an embodiment of a data transmission method provided by the present application. This method may comprise the following step 101 ~ step 108:

In step 101, a data requester terminal generates an asymmetrical key pair comprising a first public key and a first private key.

In step 102, the data requester terminal sends a data request carrying the first public key to a data provider terminal.

In step 103, the data provider terminal obtains an asymmetrical key pair comprising a second public key and a second private key.

In step 104, the data provider terminal generates a shared key based on the second private key and the first public key using a preset key-agreement algorithm.

In step 105, the data provider terminal uses the shared key to encrypt target data to which the data request corresponds.

In step 106, the data provider terminal sends a ciphertext obtained from encryption and the second public key to the data requester terminal.

5 In step 107, the data requester terminal generates a shared key based on the first private key and the second public key using the key-agreement algorithm.

In step 108, the data requester terminal uses the shared key to decrypt the ciphertext to obtain the target data.

10 Here, the data requester terminal is the terminal requesting data, and the data provider terminal is the terminal providing data. In an example, the data requester terminal may be a client, the data provider terminal may be a server, and the client requests the server to return target data. Taking target data that is a seed parameter for generating an offline payment code as an example, the data request may be a request for activating offline payment, the data requester terminal is a client, and the data provider terminal is a server. The client sends a
15 request for activating offline payment to the server, and the server returns a seed parameter to the client according to the request. In another example, the server may also request data from the client, and in this way, the data requester terminal may be a server and the data provider terminal may be a client. There is no limitation to this.

20 From the foregoing embodiment, it can be seen that an asymmetrical key pair comprising a first public key and a first private key may be generated through a data requester terminal, a data request carrying the first public key is sent to a data provider terminal, an asymmetrical key pair comprising a second public key and a second private key is obtained through the data provider terminal, a shared key is generated based on the second private key and the first public key using a preset key-agreement algorithm, then the shared key is used to encrypt target data
25 to which the data request corresponds, and lastly a ciphertext obtained from encryption and the second public key are transmitted to the data requester terminal, and the data requester terminal generates a shared key based on the first private key and the second public key using the same key-agreement algorithm. As the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key
30 generated based on the first private key and the second public key using the key-agreement algorithm, the data provider terminal may use the shared key for encryption, and the data requester terminal may use the shared key for decryption. As the key for encryption of target

data and the key for decryption of target data are identical, symmetric encryption and decryption algorithms may be used to encrypt and decrypt data. As a symmetric encryption algorithm typically conducts encryption by such means as shift cipher, while an asymmetrical encryption algorithm conducts encryption by such method as finding large prime numbers, it can be seen that the encryption process of a symmetric encryption algorithm is simpler than the encryption process of an asymmetrical encryption algorithm. Therefore, this embodiment may avoid the defect of long encryption and decryption times resulting from complex asymmetrical encryption and decryption algorithms and improve encryption and decryption efficiency. As a different random number is used each time, a different asymmetrical key pair is generated each time. Therefore, asymmetrical key pairs generated by different clients are different, too, avoiding the problem of a security risk for all client devices and server devices resulting from leakage of the key of a client device or a server device. Further, as the complete key is not exposed throughout the entire transmission process, it is meaningless even if the public key is hijacked by a hacker, thereby ensuring data security throughout the entire transmission process.

There is no specific limitation to the generation time of an asymmetrical key pair comprising a first public key and a first private key. For example, an asymmetrical key pair may be generated prior to each data request transmission. As another example, an asymmetrical key pair may be generated at a time other than right before a data request is sent, such as when other conditions are met, so that a previously generated asymmetrical key pair can be obtained when a data request is to be sent. For example, an asymmetrical key pair may be generated at set intervals, and each newly generated asymmetrical key pair replaces the previously generated asymmetrical key pair.

In an example, a first public key and a first private key may be an asymmetrical key pair generated using a key generation algorithm. Before a data request is sent, the data requester terminal uses the key generation algorithm each time to generate an asymmetrical key pair comprising a first public key and a first private key. As the asymmetrical key pair generated by the asymmetrical key algorithm is different each time under normal conditions, this may avoid the problem of leakage of a key pair stored in a fixed manner resulting in all subsequent information encrypted using the key pair being insecure.

After a data requester terminal obtains a first public key and a first private key, the data requester terminal may send a data request carrying the first public key to a data provider

terminal. Here, the data request is a request for target data.

In an example, the data requester terminal may directly include a first public key in a data request, thereby raising the speed of sending a data request.

In another example, sending a data request carrying the first public key to a data provider
5 terminal comprises:

the data requester terminal uses a private key in a requester certificate to sign the first public key to obtain first signature information. The requester certificate is a certificate issued by a designated institution to the data requester terminal.

The data requester terminal sends a data request carrying the first public key and the first
10 signature information to a data provider terminal.

The method further comprises the following steps: the data provider terminal verifies the first signature information based on a public key in the requester certificate and the first public key. If the verification is successful, the data provider terminal will send the ciphertext and the second public key to the data requester terminal.

Here, the designated institution typically refers to an institution that is authoritative and
15 can issue certificates. A certificate issued by the designated institution to a data requester terminal comprises at least a private key and a public key. In other words, a requester certificate comprises a private key and a public key.

As one signing method, a data requester terminal may use a hash algorithm to perform
20 hash operations on a first public key to obtain a first information abstract, use a private key in a requester certificate to encrypt the first information abstract to obtain first signature information, then generate a data request carrying the first public key and the first signature information based on the first signature information, and send the data request to a data provider terminal.

After the data provider terminal receives the data request, the data provider terminal may
25 verify the first signature information based on a public key in the requester certificate and the first public key. If the verification is successful, the ciphertext and the second public key will be sent to the data provider terminal.

Here, the data provider terminal may obtain the public key in the requester certificate by
30 the following method: the data requester terminal broadcasts it to the data provider terminal in

advance, or the data requester terminal sends it to the data provider terminal while sending a data request.

As one of the verification methods, a data provider terminal may use a hash algorithm to perform hash operations on a received first public key to obtain a second information abstract, use a public key in a requester certificate to decrypt first signature information to obtain a first information abstract, and verify whether the first information abstract is consistent with the second information abstract. If consistent, it means the verification is successful. The data provider terminal can execute the operation of sending a ciphertext and a second public key to the data requester terminal only after the verification is successful.

From the foregoing embodiment, it can be seen that signing a first public key and successfully verifying the first signature information may guarantee the first public key is not tampered with, and meanwhile a requester certificate ensures the data requester terminal is a safe terminal authenticated by an authoritative institution, thereby ensuring the security of the negotiation process of a shared key.

After a data provider terminal receives a data request, the data provider terminal may obtain an asymmetrical key pair comprising a second public key and a second private key. Here, the second public key and the second private key may be a key pair generated using the key generation algorithm. It can be seen that the asymmetrical key pair comprising a first public key and a first private key and the asymmetrical key pair comprising a second public key and a second private key are generated by an identical key generation algorithm. As the key generation algorithm uses a different random number during each calculation, asymmetrical key pairs generated during calculation at different times would almost always be different. Therefore, the asymmetrical key pair generated by the data requester terminal is different from the asymmetrical key pair generated by the data provider terminal under normal circumstances.

There is no specific limitation to the generation time of an asymmetrical key pair comprising a second public key and a second private key. For example, an asymmetrical key pair may be generated each time a data request is received. As another example, an asymmetrical key pair may be generated not when a data request is received but when other conditions are met so that a previously generated asymmetrical key pair can be obtained when a data request is received. For example, an asymmetrical key pair may be generated at set intervals, and each newly generated asymmetrical key pair replaces the previously generated asymmetrical key pair.

In an example, when a data request is received, a data provider terminal uses a key generation algorithm each time to generate an asymmetrical key pair comprising a second public key and a second private key. As the asymmetrical key pair generated by the asymmetrical key algorithm is different each time under normal conditions, this may avoid the problem of leakage of a key pair stored in a fixed manner resulting in all subsequent information encrypted using the key pair being insecure.

After a data provider terminal obtains an asymmetrical key pair comprising a second public key and a second private key, the data provider terminal may generate a shared key based on the second private key and the first public key using a preset key-agreement algorithm. Subsequently, a data requester terminal will generate a shared key based on a first private key and the second public key using the key-agreement algorithm.

A key-agreement algorithm, also known as a key exchange algorithm, may be an ECDH algorithm, for example. Here, ECDH is a DH (Diffie-Hellman) key exchange algorithm based on ECC (Elliptic Curve Cryptosystems). Therefore, the two parties may negotiate to obtain a common key without sharing any secret information.

In this embodiment, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm. As one implementation manner, the key-agreement algorithms used by a data provider terminal and a data requester terminal are identical, the key generation algorithms used by the data provider terminal and the data requester terminal are also identical, and the key-agreement algorithm and the key generation algorithm meet the following condition: for any two asymmetrical key pairs generated using the key generation algorithm, when a public key of any of the two asymmetrical key pairs and a private key of the other asymmetrical key pair are selected, negotiation results obtained using the key-agreement algorithm are identical.

It can be seen that, as a first public key typically is not equal to a second public key and a first private key typically is not equal to a second private key, it is impossible to deduce a first private key from a first public key or deduce a second private key from a second public key, and a shared key negotiated from a first private key and a second public key is identical to a shared key negotiated from a second private key and a first public key, the complete key is not exposed at any time during the entire transmission process, and data security is ensured throughout the whole transmission process. Further, as the key for encryption of target data and

the key for decryption of target data are identical, symmetric key encryption and decryption algorithms are used to encrypt and decrypt data to avoid long encryption and decryption times resulting from complex asymmetrical encryption and decryption algorithms, thereby improving encryption and decryption efficiency.

- 5 After a data provider terminal generates a shared key, the data provider terminal may use the shared key to encrypt target data to which the data request corresponds and send a ciphertext obtained from encryption and the second public key to the data requester terminal.

In an example, a data provider terminal may directly send a second public key to a data requester terminal to improve sending efficiency.

- 10 In another example, the method further comprises:

the data provider terminal uses a private key in a provider certificate to sign the second public key to obtain second signature information. The provider certificate is a certificate issued by a designated institution to the data provider terminal.

- 15 When the data provider terminal sends a ciphertext obtained from encryption and the second public key to the data requester terminal, the data provider terminal will further send the second signature information to the data requester terminal.

The data requester terminal verifies the second signature information based on a public key in the provider certificate and the second public key. If the verification is successful, the data requester terminal will decrypt the ciphertext.

- 20 Here, the designated institution may be an institution that can issue certificates. A certificate issued by the designated institution to a data provider terminal comprises at least a private key and a public key. In other words, a provider certificate comprises a private key and a public key.

- 25 As one signing method, a data provider terminal may use a hash algorithm to perform hash operations on a second public key to obtain a third information abstract, use a private key in a provider certificate to encrypt the third information abstract to obtain second signature information, and then send the ciphertext, second public key, and second signature information to the data requester terminal.

- 30 The data requester terminal verifies the second signature information based on a public key in the provider certificate and the second public key. If the verification is successful, the

data requester terminal will decrypt the ciphertext.

Here, the data requester terminal may obtain the public key in the provider certificate by the following method: the data provider terminal broadcasts it to the data requester terminal in advance, or the data provider terminal sends it to the data requester terminal while sending a
5 ciphertext and a second public key.

As one of the verification methods, a data requester terminal may use a hash algorithm to perform hash operations on a received second public key to obtain a fourth information abstract, use a public key in a provider certificate to decrypt the second signature information to obtain a third information abstract, and verify whether the third information abstract is
10 consistent with the fourth information abstract. If consistent, it means the verification is successful. The data requester terminal can execute the operation of decrypting a ciphertext only after the verification is successful.

From the foregoing embodiment, it can be seen that signing a second public key and successfully verifying the second signature information may guarantee the second public key
15 is not tampered with, and meanwhile a provider certificate ensures the data provider terminal is a safe terminal authenticated by an authoritative institution, thereby ensuring the security of the negotiation process of a shared key.

As shown in FIG. 2, FIG. 2 is a flow chart of another embodiment of a data transmission method provided by the present application. The embodiment applies the data transmission
20 method to transmit a seed parameter. The method is applicable in a client and may comprise the following step 201~step 203:

In step 201, generating an asymmetrical key pair comprising a first public key and a first private key, and sending a data request carrying the first public key to a server.

Here, it is not necessary to set a specific limitation to the generation time of an
25 asymmetrical key pair comprising a first public key and a first private key. For example, an asymmetrical key pair may be generated prior to each data request transmission. As another example, an asymmetrical key pair may be generated at a time other than right before a data request is sent but when other conditions are met, so that a previously generated asymmetrical key pair can be obtained when a data request is sent. For example, an asymmetrical key pair
30 may be generated at set intervals, and each newly generated asymmetrical key pair replaces the previously generated asymmetrical key pair.

In an example, a first public key and a first private key may be an asymmetrical key pair generated using a key generation algorithm. Before a data request is sent, the client uses the key generation algorithm each time to generate an asymmetrical key pair comprising a first public key and a first private key. As the asymmetrical key pair generated by the asymmetrical key algorithm is different each time under normal conditions, this may avoid the problem of leakage of a key pair stored in a fixed manner resulting in all subsequent information encrypted using the key pair being insecure.

After a first public key and a first private key are obtained, a data request carrying the first public key may be sent to a server. Here, the data request is for requesting the server to return a seed parameter for generating an offline payment code.

In an example, a first public key may be directly carried in a data request, thereby raising the speed of sending a data request.

In another example, sending a data request carrying the first public key to a server comprises:

Using a private key in a client certificate to sign the first public key to obtain first signature information, wherein the client certificate is a certificate issued by a designated institution to the client.

Sending a data request carrying the first public key and the first signature information to a server so that the server uses a public key in the client certificate and the first public key to verify the first signature information, and sends the ciphertext and the second public key to the client if verification is successful.

Here, the designated institution may be an institution that can issue certificates. A certificate issued by the designated institution to a client comprises at least a private key and a public key. In other words, a client certificate comprises a private key and a public key. The server may obtain the public key in the client certificate by the following method: the client broadcasts it to the server in advance, or the client sends it to the server while sending a data request.

The present embodiment may use a private key in a client certificate to sign a first public key. For example, the client may use a hash algorithm to perform hash operations on the first public key to obtain a first information abstract, use a private key in the client certificate to encrypt the first information abstract to obtain first signature information, and then send a data

request carrying the first public key and the first signature information to a data provider terminal.

From the foregoing embodiment, it can be seen that signing a first public key facilitates a server to verify first signature information, successful verification may guarantee the first public key is not tampered with, and meanwhile a client certificate ensures the client is a safe terminal authenticated by an authoritative institution, thereby ensuring the security of the negotiation process of a shared key.

In step 202, receiving a ciphertext and a second public key sent by the server, wherein the second public key is a public key in an asymmetrical key pair obtained by the server, the asymmetrical key pair obtained by the server further comprises a second private key, and the ciphertext is information obtained by encrypting a seed parameter for generating an offline payment code using a shared key; the shared key is a key generated based on the second private key and the first public key using a preset key-agreement algorithm. In step 203, generating a shared key based on the first private key and the second public key using the key-agreement algorithm, and using the shared key to decrypt the ciphertext and obtain the seed parameter.

Here, a key-agreement algorithm, also known as a key exchange algorithm, may be an ECDH algorithm, for example, wherein ECDH is a DH (Diffie-Hellman) key exchange algorithm based on ECC (Elliptic Curve Cryptosystems). Therefore, the two parties may negotiate a common key without sharing any secret information.

In this embodiment, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm. As one implementation manner, the key-agreement algorithms used by a data provider terminal and a data requester terminal are identical, the key generation algorithms used by the data provider terminal and the data requester terminal are also identical, and the key-agreement algorithm and the key generation algorithm meet the following condition: for any two asymmetrical key pairs generated using the key generation algorithm, when a public key of any of the two asymmetrical key pairs and a private key of the other asymmetrical key pair are selected, negotiation results obtained using the key-agreement algorithm are identical.

It can be seen that, as a first public key typically is not equal to a second public key and a first private key typically is not equal to a second private key, it is impossible to deduce a first private key from a first public key or deduce a second private key from a second public

key, and the shared key negotiated from a first private key and a second public key is identical to the shared key negotiated from a second private key and a first public key, the complete key is not exposed at any time during the entire transmission process, and data security is ensured throughout the whole transmission process. Further, as a key for encryption of a seed parameter and a key for decryption of a seed parameter are identical, symmetric key encryption and decryption algorithms are used to encrypt and decrypt a seed parameter to avoid long encryption and decryption times resulting from complex asymmetrical encryption and decryption algorithms, thereby improving encryption and decryption efficiency.

In an optional implementation manner, a client may be in an electronic device, particularly in a wearable device. As a wearable device often has configurations corresponding to relatively low performance, when this solution is used in a wearable device, symmetric encryption and decryption algorithms are not very demanding on resources. While ensuring transmission security, this solution greatly improves performance and improves the efficiency of the whole transmission process. Further, a wearable device may comprise a smart bracelet. Implementing the embodiment through a smart bracelet may not only guarantee the transmission security of a seed parameter but also ensure the efficiency of the whole transmission process.

In an example, the method in the embodiment may be executed through a secure element (SE), thereby enabling generation of an asymmetrical key, generation of a shared key, and decryption of a ciphertext to be executed in the SE. Further, a seed parameter may also be stored in an SE. As the SE has an anti-cracking function, the SE may provide a seed parameter with a very high security level.

Further, a seed parameter may be stored in an SE, and meanwhile access authority of the SE may be set, too, with payment code generation being controlled through fingerprint recognition, pulse recognition, face recognition, or other verification methods, thereby providing the whole payment code with a very high security level.

As shown in FIG. 3, FIG. 3 is a flow chart of another embodiment of a data transmission method provided by the present application. The embodiment uses the data transmission method to transmit a seed parameter. When the method is used in a server, it may comprise the following step 301 ~ step 303:

In step 301, receiving a data request carrying a first public key and sent by a client, wherein the data request is for requesting the server to return a seed parameter for generating

an offline payment code, the first public key is a public key in an asymmetrical key pair generated by the client, and the asymmetrical key pair generated by the client further comprises a first private key.

Here, when a data request is received, if the data request only carries a first public key, step 302 may be executed directly; if the data request carries a first public key and first signature information, the first signature information is verified based on a public key in a client certificate and the first public key, and step 302 is executed only after the verification is successful.

Here, the server may obtain the public key in the client certificate by the following method: the client broadcasts it to the server in advance, or the client sends it to the server while sending a data request.

As one of the verification methods, a server may use a hash algorithm to perform hash operations on a received first public key to obtain a second information abstract, use a public key in a client certificate to decrypt the first signature information to obtain a first information abstract, and verify whether the first information abstract is consistent with the second information abstract. If consistent, it means the verification is successful. The server subsequently will return a ciphertext and a second key to the client.

In step 302, obtaining an asymmetrical key pair comprising a second public key and a second private key, and generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm.

In step 303, using the shared key to encrypt a seed parameter to which the data request corresponds, and sending a ciphertext obtained from encryption and the second public key to the client so that the client generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext and obtain the seed parameter

Here, it is not necessary to set specific limitations to generation time of an asymmetrical key pair comprising a second public key and a second private key. For example, an asymmetrical key pair may be generated each time a data request is received. As another example, an asymmetrical key pair may be generated at a time other than when a data request is received but when other conditions are met so that a previously generated asymmetrical key pair can be obtained when a data request is received. For example, an asymmetrical key pair

may be generated at set intervals, and each newly generated asymmetrical key pair replaces the previously generated asymmetrical key pair.

In an example, the second public key and the second private key may be a key pair generated using the key generation algorithm. For example, when a data request is received, a server uses the key generation algorithm each time to generate an asymmetrical key pair comprising a second public key and a second private key. As the asymmetrical key pair generated by the asymmetrical key algorithm is different each time under normal conditions, this may avoid the problem of leakage of a key pair stored in a fixed manner resulting in all subsequent information encrypted using the key pair being insecure.

Here, the asymmetrical key pair comprising a first public key and a first private key and the asymmetrical key pair comprising a second public key and a second private key are generated by an identical key generation algorithm. As the key generation algorithm uses a different random number during each calculation, the asymmetrical key pairs generated during calculation at different times would seem different. Therefore, the asymmetrical key pair generated by the client is different from the asymmetrical key pair generated by the server under normal circumstances.

After a server obtains a second public key and a second private key, the server may generate a shared key based on the second private key and the first public key using a preset key-agreement algorithm.

A key-agreement algorithm, also known as a key exchange algorithm, may be an ECDH algorithm, for example, wherein ECDH is a DH (Diffie-Hellman) key exchange algorithm based on ECC (Elliptic Curve Cryptosystems). Therefore, the two parties may negotiate a common key without sharing any secret information.

In this embodiment, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm. As one implementation manner, the key-agreement algorithms used by a data provider terminal and a data requester terminal are identical, the key generation algorithms used by the data provider terminal and the data requester terminal are also identical, and the key-agreement algorithm and the key generation algorithm meet the following condition: for any two asymmetrical key pairs generated using the key generation algorithm, when a public key of any of the two asymmetrical key pairs and a private key of the other asymmetrical key

pair are selected, negotiation results obtained using the key-agreement algorithm are identical.

After a server obtains a shared key, the server may use the shared key to encrypt a seed parameter to which the data request corresponds and send a ciphertext obtained from encryption and the second public key to a client. The seed parameter is a seed parameter for generating an offline payment code. After the server receives a data request, the server may obtain a seed parameter according to the data request. The seed parameters to which clients correspond may be the same or different, subject to actual requirements.

As for the sending of a second public key, in one example, the second public key may be directly sent to a client to raise sending speed.

In another example, the method further comprises:

using a private key in a server certificate to sign the second public key to obtain second signature information. The server certificate is a certificate issued by a designated institution to the server.

While sending a ciphertext obtained from encryption and the second public key to the client, also sending the second signature information to the client, so that the client verifies the second signature information based on a public key in the server certificate and the second public key. If the verification is successful, the client will decrypt the ciphertext.

Here, the designated institution may be an institution that can issue certificates. A certificate issued by the designated institution to a server comprises at least a private key and a public key. In other words, a server certificate comprises a private key and a public key. The client may obtain the public key in the server certificate by the following method: the server broadcasts it to the client in advance, or the server sends it to the client while sending a ciphertext and a second public key.

As one signing method, a server may use a hash algorithm to perform hash operations on a second public key to obtain a third information abstract, use a private key in a server certificate to encrypt the third information abstract to obtain second signature information, and then send the ciphertext, the second public key, and the second signature information to the client.

The client may verify the second signature information based on the public key in the server certificate and the second public key. If successful, the client will decrypt the ciphertext.

As one of the verification methods, a client may use a hash algorithm to perform hash operations on a second public key to obtain a fourth information abstract, use a public key in a server certificate to decrypt second signature information to obtain a third information abstract, and verify whether the third information abstract is consistent with the fourth information abstract. If consistent, it means the verification is successful. The client can execute the operation of decrypting a ciphertext only after successful verification.

From the foregoing embodiment, it can be seen that signing a second public key and successfully verifying second signature information may guarantee the second public key is not tampered with, and meanwhile a server certificate ensures the server is a safe end authenticated by an authoritative institution, thereby ensuring the security of the negotiation process of a shared key.

Corresponding to an embodiment of a data transmission method provided by the present application, the present application further provides embodiments of a data transmission apparatus and a data transmission system.

Refer to FIG. 4. It is a block diagram of an embodiment of a data transmission system provided by the present application:

The system 40 comprises a data requester device 41 and a data provider device 42.

The data requester device 41 generates an asymmetrical key pair comprising a first public key and a first private key and sends a data request carrying the first public key to the data provider device 42.

The data provider device 42 obtains an asymmetrical key pair comprising a second public key and a second private key and generates a shared key based on the second private key and the first public key using a preset key-agreement algorithm.

The data provider device 42 uses the shared key to encrypt target data to which the data request corresponds and sends a ciphertext obtained from encryption and the second public key to the data requester device 41.

The data requester device 41 generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the target data.

Here, the shared key generated based on the second private key and the first public key

using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

In an optional implementation manner, the data requester device 41 uses a private key in a requester certificate to sign the first public key to obtain first signature information and sends
 5 a data request carrying the first public key and the first signature information to a data provider device 42. The requester certificate is a certificate issued by a designated institution to the data requester device.

Before the data provider device 42 returns a ciphertext and a second public key to the data requester device 41, the data provider device 42 verifies the first signature information
 10 based on a public key in the requester certificate and the first public key and determines the verification is successful.

In an optional implementation manner, the data provider device 42 uses a private key in a provider certificate to sign the second public key to obtain second signature information. When a ciphertext obtained from encryption and the second public key are sent to the data
 15 requester device 41, the data provider device 42 sends the second signature information to the data requester device 41, too; the provider certificate is a certificate issued by a designated institution to the data provider device 42.

Before the data requester device 41 decrypts a ciphertext, the data requester device 41 verifies the second signature information based on a public key in the provider certificate and
 20 the second public key and determines the verification is successful.

Refer to FIG. 5. It is a block diagram of an embodiment of a data transmission apparatus provided by the present application:

The apparatus comprises: a key generation module 51, a request sending module 52, an information receiving module 53, a shared key generation module 54, and an information
 25 decryption module 55.

The key generation module 51 is for generating an asymmetrical key pair comprising a first public key and a first private key.

The request sending module 52 is for sending a data request carrying the first public key to a server.

30 The information receiving module 53 is for receiving a ciphertext and a second public

key sent by the server, wherein the second public key is a public key in an asymmetrical key pair obtained by the server, the asymmetrical key pair obtained by the server further comprises a second private key, and the ciphertext is information obtained by encrypting a seed parameter for generating an offline payment code using a shared key; the shared key is a key generated
 5 based on the second private key and the first public key using a preset key-agreement algorithm.

The shared key generation module 54 is for generating a shared key based on the first private key and the second public key using the key-agreement algorithm.

The information decryption module 55 is for using the shared key to decrypt the ciphertext and obtain the seed parameter.

10 Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

In an optional implementation manner, the request sending module 52 is for:

15 using a private key in a client certificate to sign the first public key to obtain first signature information, wherein the client certificate is a certificate issued by a designated institution to the client.

20 sending a data request carrying the first public key and the first signature information to a server so that the server uses a public key in the client certificate and the first public key to verify the first signature information, and sends the ciphertext and the second public key to the client if verification is successful.

Refer to FIG. 6. It is a block diagram of another embodiment of a data transmission apparatus provided by the present application:

25 The apparatus comprises: a request receiving module 61, a key obtaining module 62, a shared key generation module 63, an information encryption module 64, and an information sending module 65.

30 Here, the request receiving module 61 is for receiving a data request carrying a first public key and sent by a client, wherein the data request is for requesting a server to return a seed parameter for generating an offline payment code, the first public key is a public key in an asymmetrical key pair generated by the client, and the asymmetrical key pair generated by the client further comprises a first private key.

The key obtaining module 62 is for obtaining an asymmetrical key pair comprising a second public key and a second private key.

The shared key generation module 63 is for generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm.

5 The information encryption module 64 is for using the shared key to encrypt a seed parameter to which the data request corresponds.

10 The information sending module 65 is for sending a ciphertext obtained from encryption and the second public key to the client so that the client generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the seed parameter.

Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

15 In an optional implementation manner, the apparatus 60 further comprises (not shown in FIG. 6):

a signature module for using a private key in a server certificate to sign the second public key to obtain second signature information, wherein the server certificate is a certificate issued by a designated institution to the server.

20 The information sending module 65 is further used to send the second signature information to the client when sending a ciphertext obtained from encryption and the second public key to the client so that the client verifies the second signature information based on a public key in the server certificate and the second public key. After the verification is successful, the client decrypts the ciphertext.

25 Based on this, the present application further provides a wearable device. The wearable device comprises an SE chip, which is for:

generating an asymmetrical key pair comprising a first public key and a first private key and sending a data request carrying the first public key to a server.

30 receiving a ciphertext and a second public key sent by the server, wherein the second public key is a public key in an asymmetrical key pair obtained by the server, the asymmetrical key pair obtained by the server further comprises a second private key, and the ciphertext is

information obtained by encrypting a seed parameter for generating an offline payment code using a shared key; the shared key is a key generated based on the second private key and the first public key using a preset key-agreement algorithm.

generating a shared key based on the first private key and the second public key using
 5 the key-agreement algorithm, and using the shared key to decrypt the ciphertext and obtain the seed parameter.

Here, the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

10 From the foregoing embodiment, it can be seen that through configuration of an SE in a wearable device, generation of an asymmetrical key pair, generation of a shared key, storage of target data, and decryption of a ciphertext are conducted in the SE. Further, as the SE has an anti-cracking function, the SE may provide the target data with a very high security level.

In order to avoid the problems of information security and prolonged encryption and
 15 decryption in current technologies, the present application further provides an alternative data transmission method, as shown in FIG. 7. FIG. 7 is a flow chart of another embodiment of a data transmission method provided by the present application. The method may comprise the following step 701 ~ step 708:

In step 701, a data requester terminal generates a first symmetric key.

20 In step 702, the data requester terminal sends a data request carrying the first symmetric key to a data provider terminal.

In step 703, the data provider terminal obtains a second symmetric key, which is different from the first symmetric key.

In step 704, the data provider terminal generates a shared key based on the first symmetric
 25 key and the second symmetric key using a preset key-agreement algorithm.

In step 705, the data provider terminal uses the shared key to encrypt target data to which the data request corresponds.

In step 706, the data provider terminal sends a ciphertext obtained from encryption and the second symmetric key to the data requester terminal.

In step 707, the data requester terminal generates a shared key based on the first symmetric key and the second symmetric key using the key-agreement algorithm.

In step 708, the data requester terminal uses the shared key to decrypt the ciphertext to obtain the target data.

5 From the foregoing embodiment, it can be seen that a first symmetric key may be obtained through a data requester terminal, a data request carrying the first symmetric key is sent to a data provider terminal, a second symmetric key is obtained through a data provider terminal, a shared key is generated based on the first symmetric key and the second symmetric key using a preset key-agreement algorithm, then the shared key is used to encrypt target data
10 to which the data request corresponds, and lastly a ciphertext obtained from encryption and the second symmetric key are transmitted to the data requester terminal, and the data requester terminal uses the same key-agreement algorithm to generate a shared key based on the first symmetric key and the second symmetric key. As a data provider terminal and a data requester terminal use an identical key-agreement algorithm, the shared key generated by the data
15 provider terminal and that generated by the data requester terminal are identical, and the data requester terminal may decrypt the ciphertext through the generated shared key, thereby obtaining the target data. It can be seen that, as the asymmetrical key algorithm uses a different random number during each calculation, a different asymmetrical key pair is generated during each calculation. Therefore, asymmetrical key pairs generated by different clients are different,
20 too, avoiding the problem of a security risk for all client devices and server devices resulting from leakage of the key of a client device or a server device. Meanwhile as the shared key is different from the first symmetric key and the second symmetric key, even if a hacker has hijacked the symmetric key, the hacker will not know what key-agreement algorithm the present application has used, so the hacker is unable to decrypt the ciphertext, thereby ensuring
25 data security throughout the entire transmission process. Further, as the key for encryption of target data and the key for decryption of target data are identical, symmetric key encryption and decryption algorithms are used to encrypt and decrypt data to avoid long encryption and decryption times resulting from complex asymmetrical encryption and decryption algorithms, thereby improving encryption and decryption efficiency.

30 Here, the key generation algorithm generating a first symmetric key and the key generation algorithm generating a second symmetric key may be the same or different. No specific limitation is set. As the symmetric key generated by the key generation algorithm is

different each time, the second symmetric key is different from the first symmetric key.

After the data requester terminal obtains a first symmetric key, it may generate a data request carrying a first symmetric key based on the first symmetric key. Here, the data request is used to request target data. After the data request is generated, it may be sent to a data provider terminal.

In an example, the data requester terminal may use a data request to directly carry a first symmetric key, thereby raising the speed of sending a data request.

In another example, sending a data request carrying the first symmetric key to a data provider terminal comprises:

the data requester terminal uses a private key in a requester certificate to sign the first symmetric key to obtain first signature information. The requester certificate is a certificate issued by a designated institution to the data requester terminal.

The data requester terminal sends a data request carrying the first symmetric key and the first signature information to a data provider terminal.

Here, the designated institution may be an institution that can issue certificates. A certificate issued by the designated institution to a data requester terminal comprises at least a private key and a public key.

After a data provider terminal receives a data request, the data provider terminal may verify the first signature information based on a public key in the requester certificate and the first symmetric key. After the verification is successful, the data provider terminal will execute the operation of returning a ciphertext and a second symmetric key to the data requester terminal.

From the foregoing embodiment, it can be seen that signing a first symmetric key and successfully verifying the first signature information may guarantee the first symmetric key is not tampered with, and meanwhile a requester certificate ensures the data requester terminal is a safe terminal authenticated by an authoritative institution, thereby ensuring the security of the negotiation process of a shared key.

After a data provider terminal receives a data request, the data provider terminal may obtain a second symmetric key. After the data provider terminal obtains the second symmetric key, the data provider terminal may generate a shared key based on the first symmetric key and

the second symmetric key using a preset key-agreement algorithm.

Here, a key-agreement algorithm, also known as a key exchange algorithm, may be an ECDH algorithm, for example, wherein ECDH is a DH (Diffie-Hellman) key exchange algorithm based on ECC (Elliptic Curve Cryptosystems). Therefore, the two parties may
5 negotiate a common key without sharing any secret information.

A data provider terminal may use the shared key to encrypt target data to which the data request corresponds and send a ciphertext obtained from encryption and the second symmetric key to the data requester terminal.

In an example, a data provider terminal may directly send a second symmetric key to a
10 data requester terminal to improve sending efficiency.

In another example, the data provider terminal uses a private key in a provider certificate to sign the second symmetric key to obtain second signature information. The provider certificate is a certificate issued by a designated institution to the data provider terminal.

When the data provider terminal sends a ciphertext obtained from encryption and the
15 second symmetric key to the data requester terminal, the data provider terminal will further send the second signature information to the data requester terminal.

Here, the designated institution may be an institution that can issue certificates. A certificate issued by the designated institution to a data provider terminal comprises at least a private key and a public key. In other words, a provider certificate comprises a private key and
20 a public key.

The data requester terminal verifies the second signature information based on a public key in the provider certificate and the second symmetric key. After the verification is successful, the data requester terminal executes the step of decrypting a ciphertext.

From the foregoing embodiment, it can be seen that signing a second symmetric key and
25 successfully verifying second signature information may guarantee the second symmetric key is not tampered with, and meanwhile a provider certificate ensures the data provider terminal is a safe terminal authenticated by an authoritative institution, thereby ensuring the security of the negotiation process of a shared key.

Corresponding to the embodiment of a data transmission method provided by the present
30 application, the present application further provides an embodiment of a data transmission

system.

Refer to FIG. 8. It is a block diagram of another embodiment of a data transmission system provided by the present application:

The system 80 comprises a data requester device 81 and a data provider device 82.

5 The data requester device 81 generates a first symmetric key and sends a data request carrying the first symmetric key to the data provider device 82.

The data provider device 82 obtains a second symmetric key and generates a shared key based on the first symmetric key and the second symmetric key using a preset key-agreement algorithm, the second symmetric key being different from the first symmetric key.

10 The data provider device 82 uses the shared key to encrypt target data to which the data request corresponds and sends a ciphertext obtained from encryption and the second symmetric key to the data requester device 81.

15 The data requester device 81 generates a shared key based on the first symmetric key and the second symmetric key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the target data.

The processes of achieving the functions and effects of every module in the foregoing apparatus are detailed in the achieving processes of corresponding steps in the foregoing method, so no unnecessary details will be repeated.

20 The apparatus embodiment basically corresponds to the method embodiment, so for relevant parts of the apparatus, please refer to the corresponding parts of the method embodiment. The apparatus embodiment described above is exemplary only, its units described as separate components may or may not be physically separate, and the components displayed as units may or may not be physical units, i.e., they may be located in the same place or distributed on a plurality of network units. Some or all of the modules may be selected
25 according to the actual need to achieve the objectives of the solution of the present application. Those of ordinary skill in the art can understand and implement it without creative effort.

30 After considering the Description and practicing the invention applied for here, those skilled in the art may easily think of other embodiments of the present application. The present application is intended to cover any modification, use, or adaptive change of the present application. These modifications, uses, or adaptive changes follow the general principle of the

present application and include the common general knowledge or conventional technical means in the technical field not applied for by the present application. The Description and embodiments are exemplary only. The real scope and spirit of the present application are stated by the following Claims.

- 5 It should be understood that the present application is not limited to the precise structure described above and shown in the accompanying drawings, and various modifications and changes may be made without departing from its scope. The scope of the present application is limited by the attached Claims.

CLAIMS

1. A data transmission method, wherein the method is applicable in a client and comprises:

generating an asymmetrical key pair comprising a first public key and a first private key,
5 and sending a data request carrying the first public key to a server;

receiving a ciphertext and a second public key sent by the server, wherein the second public key is a public key in an asymmetrical key pair obtained by the server, the asymmetrical key pair obtained by the server further comprises a second private key, and the ciphertext is information obtained by encrypting a seed parameter for generating an offline payment code
10 using a shared key; the shared key is a key generated based on the second private key and the first public key using a preset key-agreement algorithm;

generating a shared key based on the first private key and the second public key using the key-agreement algorithm, and using the shared key to decrypt the ciphertext to obtain the seed parameter;

15 wherein the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

2. The method as described in claim 1, wherein the sending a data request carrying the first public key to a server comprises:

20 using a private key in a client certificate to sign the first public key to obtain first signature information, wherein the client certificate is a certificate issued by a designated institution to the client;

sending a data request carrying the first public key and the first signature information to a server so that the server uses a public key in the client certificate and the first public key to
25 verify the first signature information, and sends the ciphertext and the second public key to the client if verification is successful.

3. The method as described in claim 1 or 2, wherein the client is located in a wearable device.

4. The method as described in claim 3, wherein the wearable device comprises a smart
30 bracelet.

5. A data transmission method, wherein the method is applicable in a server and comprises:

receiving a data request carrying a first public key and sent by a client, wherein the data request is for requesting the server to return a seed parameter for generating an offline payment code, the first public key is a public key in an asymmetrical key pair generated by the client, and the asymmetrical key pair generated by the client further comprises a first private key;

obtaining an asymmetrical key pair comprising a second public key and a second private key, and generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

using the shared key to encrypt a seed parameter to which the data request corresponds, and sending a ciphertext obtained from encryption and the second public key to the client so that the client generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the seed parameter;

wherein the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

6. The method as described in claim 5, wherein the method further comprises:

using a private key in a server certificate to sign the second public key to obtain second signature information, wherein the server certificate is a certificate issued by a designated institution to the server;

while sending the ciphertext obtained from encryption and the second public key to the client, also sending the second signature information to the client, so that the client verifies the second signature information based on a public key in the server certificate and the second public key, and decrypts the ciphertext if verification is successful.

7. A data transmission method, wherein the method comprises:

a data requester terminal generating an asymmetrical key pair comprising a first public key and a first private key and sending a data request carrying the first public key to a data provider terminal;

the data provider terminal obtaining an asymmetrical key pair comprising a second public

key and a second private key and generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

the data provider terminal using the shared key to encrypt target data to which the data request corresponds and sending a ciphertext obtained from encryption and the second public key to the data requester terminal;

the data requester terminal generating a shared key based on the first private key and the second public key using the key-agreement algorithm and using the shared key to decrypt the ciphertext to obtain the target data;

wherein the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

8. The method as described in claim 7, wherein the sending a data request carrying the first public key to a data provider terminal comprises:

the data requester terminal using a private key in a requester certificate to sign the first public key to obtain first signature information, wherein the requester certificate is a certificate issued by a designated institution to the data requester terminal;

the data requester terminal sending a data request carrying the first public key and the first signature information to the data provider terminal;

the method further comprising: the data provider terminal verifying the first signature information based on a public key in the requester certificate and the first public key, and the data provider terminal sending the ciphertext and the second public key to the data requester terminal if verification is successful.

9. The method as described in claim 7, wherein the method further comprises:

the data provider terminal using a private key in a provider certificate to sign the second public key to obtain second signature information, wherein the provider certificate is a certificate issued by a designated institution to the data provider terminal;

when the data provider terminal sends the ciphertext obtained from encryption and the second public key to the data requester terminal, the data provider terminal further sending the second signature information to the data requester terminal;

the data requester terminal verifying the second signature information based on a public key in the provider certificate and the second public key, and the data requester terminal decrypting the ciphertext if verification is successful.

10. A data transmission apparatus, wherein the apparatus comprises:

5 a key generation module for generating an asymmetrical key pair comprising a first public key and a first private key;

a request sending module for sending a data request carrying the first public key to a server;

10 an information receiving module for receiving a ciphertext and a second public key sent by the server, wherein the second public key is a public key in an asymmetrical key pair obtained by the server, the asymmetrical key pair obtained by the server further comprises a second private key, and the ciphertext is information obtained by encrypting a seed parameter for generating an offline payment code using a shared key; the shared key is a key generated based on the second private key and the first public key using a preset key-agreement algorithm;

15 a shared key generation module for generating a shared key based on the first private key and the second public key using the key-agreement algorithm;

an information decryption module for using the shared key to decrypt the ciphertext to obtain the seed parameter;

20 wherein the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

11. A data transmission apparatus, wherein the apparatus comprises:

25 a request receiving module for receiving a data request carrying a first public key and sent by a client, wherein the data request is for requesting a server to return a seed parameter for generating an offline payment code, the first public key is a public key in an asymmetrical key pair generated by the client, and the asymmetrical key pair generated by the client further comprises a first private key;

a key obtaining module for obtaining an asymmetrical key pair comprising a second public key and a second private key;

a shared key generation module for generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

an information encryption module for using the shared key to encrypt a seed parameter to which the data request corresponds,

5 an information sending module for sending a ciphertext obtained from encryption and the second public key to the client so that the client generates a shared key based on the first private key and the second public key using the key-agreement algorithm and uses the shared key to decrypt the ciphertext to obtain the seed parameter;

10 wherein the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

12. A data transmission system, wherein the system comprises a data requester device and a data provider device;

15 the data requester device generating an asymmetrical key pair comprising a first public key and a first private key and sending a data request carrying the first public key to a data provider device;

the data provider device obtaining an asymmetrical key pair comprising a second public key and a second private key and generating a shared key based on the second private key and the first public key using a preset key-agreement algorithm;

20 the data provider device using the shared key to encrypt target data to which the data request corresponds and sending a ciphertext obtained from encryption and the second public key to the data requester device;

25 the data requester device generating a shared key based on the first private key and the second public key using the key-agreement algorithm and using the shared key to decrypt the ciphertext to obtain the target data.

wherein the shared key generated based on the second private key and the first public key using the preset key-agreement algorithm is identical to the shared key generated based on the first private key and the second public key using the key-agreement algorithm.

13. A data transmission method, wherein the method comprises:

a data requester terminal generating a first symmetric key and sending a data request carrying the first symmetric key to a data provider terminal;

the data provider terminal obtaining a second symmetric key and generating a shared key based on the first symmetric key and the second symmetric key using a preset key-agreement algorithm, the second symmetric key being different from the first symmetric key;

the data provider terminal using the shared key to encrypt target data to which the data request corresponds and sending a ciphertext obtained from encryption and the second symmetric key to the data requester terminal; and

the data requester terminal generating a shared key based on the first symmetric key and the second symmetric key using the key-agreement algorithm and using the shared key to decrypt the ciphertext to obtain the target data.

14. A data transmission system, wherein the system comprises a data requester device and a data provider device;

the data requester device generating a first symmetric key and sending a data request carrying the first symmetric key to the data provider device;

the data provider device obtaining a second symmetric key and generating a shared key based on the first symmetric key and the second symmetric key using a preset key-agreement algorithm, the second symmetric key being different from the first symmetric key.

the data provider device using the shared key to encrypt target data to which the data request corresponds and sending a ciphertext obtained from encryption and the second symmetric key to the data requester device;

the data requester device generating a shared key based on the first symmetric key and the second symmetric key using the key-agreement algorithm and using the shared key to decrypt the ciphertext to obtain the target data.