MESSAGE SHARING METHOD AND APPARATUS, ELECTRONIC DEVICE, AND STORAGE MEDIUM

RELATED APPLICATION

[0001] This application claims priority to Chinese Patent Application No. 2024108061543, entitled "MESSAGE SHARING METHOD AND APPARATUS, ELECTRONIC DEVICE, AND STORAGE MEDIUM" filed with the China National Intellectual Property Administration on June 20, 2024, which is incorporated herein by reference in its entirety.

FIELD OF THE TECHNOLOGY

[0002] This application relates to the field of Internet technologies, and in particular, to a message sharing method and apparatus, an electronic device, and a storage medium.

BACKGROUND OF THE DISCLOSURE

[0003] With the development of computer technologies, an increasing number of application programs have been developed and used, and different application software can share message content with each other. Currently, for different application software, message sharing may be directly performed through inter-process communication.

[0004] However, for messages with privacy, such as conversation messages involving privacy, directly transmitting messages between different application software cannot ensure effectiveness and security of message sharing.

SUMMARY

[0005] An objective of embodiments of this application is to provide a message sharing method and apparatus, and an electronic device, to resolve the problem of security of message transmission between different applications. The technical solutions provided in this application are as follows:

[0006] According to an aspect, an embodiment of this application provides a message sharing method, applied to a first server. The method includes:

receiving a message encryption request transmitted by a first application program of a first terminal, the message encryption request including message data;

encrypting the message data to obtain encrypted data;

transmitting the encrypted data to the first application program, to cause the first application program to transmit an object identifier of a sharee and the encrypted data to a second application program of the first terminal, the first application program being different from the second application program;

receiving a message decryption request transmitted by the second application program of the first terminal; and

returning decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee corresponding to the object identifier.

[0007] According to another aspect, an embodiment of this application provides a message sharing method, applied to a first application program of a first terminal, and including:

transmitting a message encryption request to a first server in response to a message sharing operation, the message encryption request including message data;

receiving encrypted data that is returned by the first server in response to the message encryption request, the encrypted data being obtained by the first server encrypting the message data; and

transmitting the encrypted data and an object identifier of a sharee to a second application program, to cause the second application program to obtain decrypted message data through the first server based on the encrypted data, and share, through a second server, the message data to the sharee corresponding to the object identifier, the first application program being different from the second application program.

[0008] According to another aspect, an embodiment of this application provides a message sharing apparatus. The apparatus includes:

a first receiving module, configured to receive a message encryption request transmitted by a first application program of a first terminal, the message encryption request including message data;

a first encryption module, configured to encrypt the message data to obtain encrypted data;

a first transmission module, configured to transmit the encrypted data to the first application program, to cause the first application program to transmit an object identifier of a

sharee and the encrypted data to a second application program of the first terminal, the first application program being different from the second application program; and

a first sharing module, configured to receive a message decryption request transmitted by a second application program of the first terminal; and return decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee corresponding to the object identifier.

[0009] According to another aspect, an embodiment of this application provides a message sharing apparatus. The apparatus includes:

a second transmission module, configured to transmit a message encryption request to a first server in response to a message sharing operation, the message encryption request including message data;

a second receiving module, configured to receive encrypted data that is returned by the first server in response to the message encryption request, the encrypted data being obtained by the first server encrypting the message data; and

a second sharing module, configured to transmit the encrypted data and an object identifier of a sharee to a second application program, to cause the second application program to obtain decrypted message data through the first server based on the encrypted data, and share, through a second server, the message data to the sharee corresponding to the object identifier, the first application program being different from the second application program.

[0010] According to another aspect, an embodiment of this application further provides an electronic device. The electronic device includes a memory and a processor. The memory has a computer program stored therein, and the processor executes the computer program to implement the method provided in any one exemplary embodiment of this application.

[0011] According to another aspect, an embodiment of this application further provides a computer-readable storage medium. The storage medium has a computer program stored therein, and the computer program, when executed by a processor, implements the method provided in any one exemplary embodiment of this application.

[0012] According to another aspect, an embodiment of this application further provides a computer program product. The computer product includes a computer program. The computer program, when executed by a processor, implements the method provided in any exemplary embodiment of this application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] To describe the technical solutions in the embodiments of this application more clearly, the accompanying drawings required for use in description of the embodiments of this application are briefly described below.

[0014] FIG. 1 is a schematic diagram of an application environment of a message sharing method according to an example.

[0015] FIG. 2 is a schematic flowchart of a message sharing method according to an embodiment of this application.

[0016] FIG. 3 is a schematic interface diagram of a message sharing page according to an example of this application.

[0017] FIG. 4 is a schematic interface diagram of a sharing page according to an example of this application.

[0018] FIG. 5 is a schematic diagram of a message sharing scheme according to an example of this application.

[0019] FIG. 6 is a schematic diagram of a message sharing scheme according to an example of this application.

[0020] FIG. 7 is a schematic flowchart of a message sharing method according to an embodiment of this application.

[0021] FIG. 8 is a schematic structural diagram of a message sharing apparatus according to an embodiment of this application.

[0022] FIG. 9 is a schematic structural diagram of a message sharing apparatus according to an embodiment of this application.

[0023] FIG. 10 is a schematic structural diagram of an electronic device according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0024] Embodiments of this application are described below with reference to the accompanying drawings of this application. The following implementations described with reference to the accompanying drawings are exemplary descriptions for explaining the technical solutions of the embodiments of this application, and are not intended to limit the technical solutions of the embodiments of this application.

[0025] Unless specifically stated, the singular forms "a", "an", "the", and "this" used herein may include plural forms. The terms "include" and "comprise" used in the embodiments of this application mean that corresponding features may be implemented as presented features, information, data, steps, operations, elements, and/or components, but do not exclude implementation as other features, information, data, steps, operations, elements, components, and/or a combination thereof that are supported in the technical field. One element is referred to as being "connected" or "coupled" to another element, the element may be directly connected or coupled to the another element, or may refer to a connection relationship established between the element and the another element through an intermediate element. In addition, the "connection" or "coupling" used herein may include wireless connection or wireless coupling. The term "and/or" used herein indicates at least one of the items defined by the term. For example, "A and/or B" indicates an implementation as "A", an implementation as "B", or an implementation as "A and B". When a plurality of (two or more) items are described, if a relationship among the plurality of items is not clearly limited, the plurality of items may refer to one, more, or all of the plurality of items. For example, the description of "a parameter A includes A1, A2, and A3" may be implemented as that the parameter A includes A1, or A2, or A3, or may be implemented as that the parameter A includes at least two of the three parameters A1, A2, and A3.

[0026] In the embodiments of this application, the term "module" or "unit" refers to a computer program or a part of a computer program with a predetermined function, which works together with other related parts to achieve a predetermined objective, and may be fully or partially implemented using software, hardware (such as a processing circuit or a memory), or a combination thereof. Similarly, one processor (or a plurality of processor or memories) may be configured to implement one or more modules or units. Furthermore, the module or unit may be a part of an integrated module or unit that includes the functionality of the module or unit.

[0027] The solutions provided in the embodiments of this application relate to technologies such as message sharing, which is specifically described through the following embodiments.

[0028] The technical solutions of this application and technical effects achieved by the technical solutions of this application are described below through description of some exemplary implementations. The following implementations may be referenced to each other or combined with each other. For same terms, similar feature, similar implementation steps, and the like in different implementations, repeated description is not be given.

[0029] In specific implementations of this application, data related to an object, such as message data, is involved. When the embodiments of this application are applied to a specific

product or technology, permission or consent from the object is required, and collection, use, and processing of relevant data need to comply with the relevant laws, regulations, and standards of the relevant countries and regions. That is, in the embodiments of this application, if the data related to the object is involved, the data needs to be obtained with the authorization and consent of the object and in compliance with the relevant laws, regulations, and standards of the countries and regions.

[0030] A message sharing method provided in the embodiments of this application may be implemented by any computer device. In an embodiment, the message sharing method may be implemented by a server. The server may be an independent physical server, or a server cluster or distributed system composed of a plurality of physical servers, or a cloud server providing a cloud computing service.

[0031] FIG. 1 is a schematic diagram of an application environment of a message sharing method according to an embodiment of this application. The application environment may include a first server 101, a first terminal 102, a second server 103, and at least one second terminal 104. The first terminal 102 includes a first client and a second client. The first client is a client of a first application program; and the second client is a client of a second application program. The second terminal 104 also includes the second client. Specifically, the first server 101 receives a message encryption request transmitted by the first client of the first terminal 102, where the message encryption request includes to-be-shared message data. The first server 101 encrypts the message data to obtain encrypted data. The first server 101 transmits the encrypted data to the first client, and the first client transmits an object identifier of a sharee and the encrypted data to the second client of the first terminal. The first server 101 receives a message decryption request transmitted by the second client of the first terminal, and returns decrypted data to the second client in response to the message decryption request, and the second client shares, through the second server 103 based on the decrypted data, decrypted message data to the sharee corresponding to the object identifier. The sharee may include the second client in the first terminal and the second client in the second terminal.

[0032] The server may be an independent physical server, or a server cluster or distributed system composed of a plurality of physical servers, or a cloud server that provides a basic cloud computing service such as a cloud service, a cloud database, cloud computing, a cloud function, cloud storage, a network service, cloud communication, a middle-ware service, a domain name service, a security service, a content delivery network (CDN), and a big data and artificial intelligence platform. The terminal may be a smartphone (such as an Android mobile phone or an iOS mobile phone), a tablet computer, a notebook computer, a digital broadcast receiver, a mobile

Internet device (MID), a personal digital assistant (PDA), a desktop computer, a smart appliance, an in-vehicle terminal (such as an in-vehicle navigation terminal or an in-vehicle computer), a smart speaker, a smartwatch, or the like. The terminal and the server may be, but are not limited to, directly or indirectly connected by using a wired or wireless communication protocol. The embodiments of this application are applicable to various scenarios, including but not limited to, a cloud technology, artificial intelligence, smart transportation, and assisted driving. Specifically, it may be determined according to an actual application scenario requirement. This is not limited herein.

[0033] The terminal (which may alternatively be referred to as a user terminal or user equipment) may be, but is not limited to, a smartphone, a tablet computer, a notebook computer, a desktop computer, a smart speech interaction device (such as a smart speaker), a wearable electronic device (such as a smartwatch), an in-vehicle terminal, a smart appliance (such as a smart television), an augmented reality (AR)/virtual reality (VR) device, or the like. The embodiments of this application are applicable to various scenarios, including but not limited to, a cloud technology, artificial intelligence, smart transportation, and assisted driving.

[0034] FIG. 2 is a schematic flowchart of a message sharing method according to an embodiment of this application. An example in which the message sharing method is performed by a first server is used. The message sharing method provided in this application may include the following operations:

[0035] Operation S201: Receive a message encryption request transmitted by a first application program of a first terminal.

[0036] The message encryption request includes message data.

[0037] In some embodiments, in operation S201, the message encryption request transmitted by the first client of the first terminal may be received. The first client is a client of the first application program, and a sharee of the message data is an object of a second application program.

[0038] The sharee is the object of the second application program, but a terminal that is used by the sharee and that corresponds to the second application program is not necessarily the first terminal.

[0039] For example, a sharer shares, through an application program A on a terminal, a message to a sharee in an application program B of the first terminal. In this case, when receiving the message, the sharee receives and views the message through the terminal of the sharee.

[0040] The first application program and the second application program are not limited to a chat application program, but both the first application program and the second application program include a function of transmitting a message between different objects.

[0041] Specifically, the message data may be determined through a selection operation of the sharer in the client of the first application program.

[0042] In a specific implementation, the first client may present a shared content selection page, and the sharer selects the to-be-shared message data through the shared content selection page, and then selects to share the message data. As shown in FIG. 3, on a message sharing page 301, a sharer may select, through the first client, to share a message to a sharee in the first application program, or may select to share a message to a sharee in the second application program, that is, forward the message to an application program B shown in the figure. If the sharer selects to share the message to the second application program, the first client automatically generates the message encryption request based on an action of the sharer selecting to share the message, and transmits the message encryption request to the first server.

[0043] If the sharer selects to share the message to the sharee in the first application program, message encryption is not required.

[0044] Operation S202: Encrypt message data to obtain encrypted data.

[0045] Specifically, different message data may be encrypted using different encryption methods. For example, encryption may be performed using a method such as key hashing, symmetric encryption, asymmetric encryption, or a digital signature.

[0046] Different message data may alternatively be encrypted using the same encryption method, but correspond to different encryption information. For example, different message data is encrypted using a key method, and different message data corresponds to different keys.

[0047] Specifically, a message data encryption process is further described in detail below.

[0048] In a specific implementation, before the message data is encrypted, validity of a login state of the first application program and correctness of a message source of the message data may be verified first. That is, after determining that the first application program is in the login state and the message source of the message data is the first application program, the first server performs message encryption.

[0049] Operation S203: Transmit the encrypted data to the first application program, to cause the first application program to transmit an object identifier of a sharee and the encrypted data to a second application program of the first terminal.

[0050] Specifically, the encrypted data may be transmitted to the second client, and the second client is a client of the second application program.

[0051] Specifically, the first application program transmitted the encrypted message to the second application program on the same first terminal. The encrypted message may be transmitted through inter-process communication on the first terminal.

[0052] In a specific implementation, the first application program transmits the encrypted data to the second application program. This can avoid tampering of the message during transmission and effectively ensure security of message transmission.

[0053] Operation S204: Receive a message decryption request transmitted by the second application program of the first terminal.

[0054] Operation S205: Return decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee corresponding to the object identifier.

[0055] The decryption request may include the encrypted data. In this case, the first server may decrypt the encrypted data to obtain the decrypted message data. Alternatively, the decryption request may include identification information only, which indicates an identifier corresponding to the message data and an identifier corresponding to an encryption method. In this way, the first server may transmit a corresponding decryption method to the second application program, and the second application program performs decryption using the decryption method. A specific process of returning the decrypted data is further described in detail below.

[0056] Specifically, after the second application program of the first terminal obtains the decrypted message data, a sharer may confirm sharing on the second application program of the first terminal. In this case, the second application program shares the message data to a sharee of the second application program through the second server corresponding to the second application program.

[0057] As shown in FIG. 4, a sharing page 401 is an interface of the second application program on the first terminal of the sharer. If the sharer confirms sharing through the interface of the second application program, the sharer may select the sharee on the interface of the second application program, and then share the message data to the sharee of the second application program, such as a user a shown in the figure.

[0058] According to the message sharing method, when intending to share a message to the

sharee of the second application program, the first application program of the first terminal transmits the message encryption request to the first server, encrypts the message data through the first server to obtain the encrypted data, and receives the encrypted data transmitted by the first server; and then transmits the encrypted data to the second application program of the first terminal. The second application program transmits the message decryption request to the first server, and obtains the decrypted message data through the first server. In this way, during message sharing between the first application program and the second application program, the encrypted data is transmitted, so that security of message transmission between different application programs can be effectively ensured.

[0059] The foregoing embodiment is further described below with reference to an example.

[0060] As shown in FIG. 5, in an example, a message sharing method of this application may include the following operations:

[0061] A first client (namely, an application program A shown in the figure) transmits a message encryption request to a first server (namely, a backend service C shown in the figure), where the message encryption request includes to-be-shared message data.

[0062] The first server encrypts the message data to obtain encrypted data.

[0063] The first server transmits the encrypted data to the first client.

[0064] The first client transmits the encrypted data to a second client (namely, an application program B shown in the figure) of a first terminal.

[0065] The second client transmits a message decryption request to the first server, where the message decryption request includes the encrypted data.

[0066] The first server decrypts the encrypted data to obtain decrypted data, namely, decrypted message data.

[0067] The first server transmits the decrypted data to the second client.

[0068] The second client shares the message data to a sharee through a second server.

[0069] In some possible implementations, operation S202 of encrypting the message data to obtain encrypted data may include:

selecting an encryption method for the message data from a plurality of candidate encryption methods; and encrypting the message data using the selected encryption method, to obtain the encrypted data.

[0070] The encrypted data further includes identification information. The identification information indicates an identifier corresponding to the message data and an identifier corresponding to the encryption method.

[0071] Specifically, the first server may be configured to encrypt messages transmitted by a plurality of different application programs, and different messages may correspond to different encryption methods. Therefore, different messages and encryption methods need to be marked with identification information. In this way, when the message decryption request is received, a corresponding decryption method can be determined for a corresponding encrypted message.

[0072] The message decryption request includes the identification information. The returning decrypted data to the second application program in response to the message decryption request includes:

querying a plurality of candidate decryption methods for a decryption method corresponding to the identification information; and

returning the decrypted data to the second application program based on the decryption method.

[0073] Specifically, a plurality of candidate encryption methods may be pre-stored in the first server. When the message encryption request is received, one encryption method may be randomly selected to encrypt the message data, and the identifier of the message data and the identifier of the selected encryption method are stored in association. When the message decryption request is received, the corresponding encryption method may be determined based on the identification information carried in the message decryption request, to determine the corresponding decryption method. The first server may decrypt the encrypted data using the decryption method, or may transmit the decryption method to the second application program, and the second application program decrypts the encrypted data using the decryption method.

[0074] A specific message encryption process is described in further detail below with reference to an embodiment.

[0075] In some possible implementations, the candidate encryption methods correspond to a plurality of security levels. The message encryption request further includes at least one of a first identifier of the first application program, a second identifier of the second application program, and security level selection information for the message data.

[0076] Specifically, the selecting an encryption method for the message data from a plurality of candidate encryption methods may include:

(1) selecting, based on at least one of a first application type of the first application program, a second application type corresponding to the second identifier, and the security level selection information for the message data, a security level from the plurality of security levels; and

(2) determining an encryption method corresponding to the selected security level from the plurality of candidate encryption methods.

[0077] Specifically, correspondences among different sharer application programs (namely, the first application programs), different sharee application programs (namely, the second application programs), and the security levels may be preset. Alternatively, the sharer provides the security level selection information, that is, the sharer may select a target security level, and the first server may determine the corresponding encryption method based on the determined target security level or the target security level selected by the sharer.

[0078] In some possible implementations, the message encryption request further includes the first identifier of the first application program and the second identifier of the second application program.

[0079] The encrypting message data to obtain encrypted data includes:

determining a first message format of the first application program that corresponds to the first identifier, and determining a second message format of the second application program that corresponds to the second identifier; and

directly encrypting the message data to obtain the encrypted data if the first message format is the same as the second message format.

[0080] Specifically, if the first message format is the same as the second message format, format conversion does not need to be performed on the message data, the message data may be directly encrypted, and the decrypted message data may be directly shared in the second application program.

[0081] In some possible implementations, the encrypting message data to obtain encrypted data further includes:

converting a data format of the message data according to the second message format if the first message format is different from the second message format; and

encrypting the message data in the converted data format, to obtain the encrypted data.

[0082] Specifically, if the first message format is different from the second message format,

the message data needs to be converted according to the second message format. In this way, the message data in the converted format can be displayed in the second application program and viewed by the sharee.

[0083] In the foregoing embodiment, the first message format of the first application program is compared with the second message format of the second application program, and if the first message format is the same as the second message format, the message data is directly encrypted to obtain the encrypted data. If the first message format is different from the second message format, the data format of the message data is converted according to the second message format, and the message data in the converted data format is encrypted, to obtain the encrypted data, so that the sharee can normally view the message data through the second application program.

[0084] In some possible implementations, the message encryption request further includes the second identifier of the second application program.

[0085] The encrypting message data to obtain encrypted data includes:

determining the second message format of the second application type that corresponds to the second identifier;

converting the data format of the message data according to the second message format; and encrypting the message data in the converted data format, to obtain the encrypted data. Specifically, alternatively, the message data may be directly converted into a target format corresponding to the second message format (that is, the data format of the message data is converted according to the second message format) without determining whether the first message format of the first application program is the same as the second message format of the second application program. This can ensure that the sharee can normally view the message data through the second application program.

[0086] In some possible implementations, the encrypting message data to obtain encrypted data includes:

converting the message data into a preset message format; and

encrypting the message data in the preset message format, to obtain the encrypted data.

[0087] The preset message format may be a universal format applicable to all application programs.

[0088] Specifically, alternatively, the data format of the message data may be directly

converted into the universal preset message format without determining whether the first message format of the first application program is the same as the second message format of the second application program nor without the need for determining the second message format of the second application program. This can omit the process of determining the first message format or the second message format while ensuring that the sharee can normally view the message data through the second application program, thereby improving message format conversion efficiency.

[0089] A specific decryption process of the encrypted data is described in detail below with reference to an embodiment.

[0090] In some possible implementations, the message decryption request includes the encrypted data.

[0091] The returning decrypted data to the second application program in response to the message decryption request may include:

decrypting the encrypted data in response to the message decryption request, to obtain the decrypted message data; and

returning the decrypted message data as the decrypted data to the second application program.

[0092] That is, the first server may receive the encrypted data transmitted by the second application program. The first server directly decrypts the encrypted data to obtain the decrypted message data, and then returns the message data to the second application program.

[0093] In some possible implementations, the message decryption request includes the identification information. The identification information indicates the identifier corresponding to the message data and the identifier corresponding to the encryption method for the message data.

[0094] The returning decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee includes:

querying for a decryption method corresponding to the identification information; and

transmitting the decryption method as the decrypted data to the second application program, to cause the second application program to decrypt the encrypted data using the decryption method, to obtain the decrypted message data, and share the message data to the sharee through the second server.

[0095] That is, alternatively, the first server may receive the first identifier transmitted by the second application program. The first server queries for the decryption method corresponding to the first identifier, and returns the decryption method to the second application program, and the second application program decrypts the encrypted data using the decryption method, to obtain the decrypted message data.

[0096] In some possible implementations, the encrypted data and the object identifier are transmitted to the second application program by the first application program using the following method:

[0097] (1) Transmit a first verification request to the second application program.

[0098] The first verification request includes the first identifier of the first application program.

[0099] Specifically, the first application program starts the second application program through an operating system interface of the first terminal, and transmits the first verification request to the second application program.

[00100] (2) Receive a second verification request transmitted by the second application program.

[00101] The second verification request includes the second identifier of the second application program. The second verification request is transmitted to the first application program after the first application program passes first validity verification performed by the second application program based on the first identifier.

[00102] (3) Perform second validity verification on the second application program based on the second identifier, and transmit the encrypted data and the object identifier to the second application program if the second application program passes the second validity verification.

[00103] Specifically, before the first application program transmits the encrypted data to the second application program, the first application program and the second application program need to perform mutual verification.

[00104] In a specific implementation, the first application program first transmits the first verification request to the second application program, to transmit the first identifier of the first application program to the second application program. The second application program verifies validity of the first application program based on the first identifier. If the first application program passes the first validity verification, the second application program transmits the second identifier of the second application program to the first application program. Similarly, the first application

program verifies validity of the second application program based on the second identifier. If the second application program also passes the second validity verification, secure communication may be established between the first application program and the second application program. That is, the two application programs successfully handshake. Then, the first application program transmits the encrypted data and the to-be-shared object identifier to the second application program through inter-process communication.

[00105] To describe the foregoing message sharing method more clearly, the message sharing method is further described with reference to an example.

[00106] As shown in FIG. 6, in an example, a message sharing method of this application may include the following operations:

[00107] A first client (namely, an application A shown in the figure) transmits a message encryption request to a first server (namely, a backend C shown in the figure), where the message encryption request includes to-be-shared message data.

[00108] The first server verifies validity of a login state of a first application program in the first client and correctness of message content.

[00109] If the first application program passes the verification, the first server converts the message data into a preset message format and encrypts the message data in the preset message format to obtain encrypted data.

[00110] The first server transmits the encrypted data to the first client.

[00111] The first client transmits the first identifier of the first application program to a second client, and the second client performs validity verification on the first application program.

[00112] If the first application program passes the validity verification, the second client transmits a second identifier of the second application program to the first client.

[00113] The first client performs validity verification on the second application program.

[00114] If the second application program also passes the validity verification, the first client transmits the encrypted data to the second client (namely, an application B shown in the figure) of a first terminal.

[00115] The second client transmits a message decryption request to the first server, where the message decryption request includes the encrypted data.

[00116] The first server decrypts the encrypted data to obtain decrypted data, namely, decrypted message data.

[00117] The first server transmits the decrypted data to the second client.

[00118] The second client shares the message data to a sharee through a second server.

[00119] In the foregoing process, if the first application program fails to pass either of verification of validity of the login state of the first application program and correctness of the message content and validity verification of the first application, or the second application program fails to pass validity verification of the second application, message sharing may be terminated.

[00120] According to the message sharing method, when intending to share a message to the sharee of the second application program, the first application program of the first terminal transmits the message encryption request to the first server, encrypts the message data through the first server to obtain the encrypted data, and receives the encrypted data transmitted by the first server; and then transmits the encrypted data to the second application program of the first terminal. The second application program transmits the message decryption request to the first server, and obtains the decrypted message data through the first server. In this way, during message sharing between the first application program and the second application program, the encrypted data is transmitted, so that security of message transmission between different application programs can be effectively ensured.

[00121] Further, the first message format of the first application program is compared with the second message format of the second application program, and if the first message format is the same as the second message format, the message data is directly encrypted to obtain the encrypted data. If the first message format is different from the second message format, the message data is converted into the target format corresponding to the second message format, and then the message data in the target format is encrypted to obtain the encrypted data, so that the sharee can normally view the message data through the second application program.

[00122] Further, alternatively, the message data may be directly converted into the target format corresponding to the second message format without determining whether the first message format of the first application program is the same as the second message format of the second application program. This can ensure that the sharee can normally view the message data through the second application program.

[00123] Further, alternatively, the message data may be directly converted into the universal preset message format without determining whether the first message format of the first application program is the same as the second message format of the second application program and without the need for determining the second message format of the second application program. This can

omit the process of determining the first message format or the second message format while ensuring that the sharee can normally view the message data through the second application program, thereby improving message format conversion efficiency.

[00124] As shown in FIG. 7, in some possible implementations, a message sharing method is provided, applied to a first application program of a first terminal and includes:

[00125] Operation S701: Transmit a message encryption request to a first server in response to a message sharing operation.

[00126] The message encryption request includes to-be-shared message data. A first client is a client of the first application program, and a sharer of the message data is an object of a second application program. The first application program is different from the second application program.

[00127] Specifically, the sharer is the object of the second application program, but a terminal that is used by the sharer and that corresponds to the second application program is not necessarily the first terminal.

[00128] The first application program and the second application program are not limited to a chat application program, but both the first application program and the second application program include a function of transmitting a message between different objects.

[00129] Operation S702: Receive encrypted data that is returned by the first server in response to the message encryption request.

[00130] The encrypted data is obtained by the first server encrypting the message data.

[00131] Specifically, different message data may be encrypted using different encryption methods. For example, encryption may be performed using a method such as key hashing, symmetric encryption, asymmetric encryption, or a digital signature.

[00132] Different message data may alternatively be encrypted using the same encryption method, but correspond to different encryption information. For example, different message data is encrypted using a key method, and different message data corresponds to different keys.

[00133] In a specific implementation, before encrypting the message data, the first server may first verify validity of a login state of the first application program and correctness of a message source of the message data. That is, after determining that the first application program is in the login state and the message source of the message data is the first application program, and then performs message encryption, the first server performs message encryption.

[00134] Operation S703: Transmit the encrypted data and an object identifier of a sharee to a second application program, to cause the second application program to obtain decrypted message data through the first server based on the encrypted data, and share, through a second server, the message data to the sharee corresponding to the object identifier.

[00135] The encrypted data and the object identifier of the sharee may be transmitted to the second client, and the second client is a client of the second application program.

[00136] Specifically, the first application program transmitted the encrypted message to the second application program on the same first terminal. The encrypted message may be transmitted through inter-process communication on the first terminal.

[00137] In a specific implementation, the first application program transmits the encrypted data to the second application program. This can avoid tampering of the message during transmission and effectively ensure security of message transmission.

[00138] According to the message sharing method, when intending to share a message to the sharee of the second application program, the first application program of the first terminal transmits the message encryption request to the first server, encrypts the message data through the first server to obtain the encrypted data, and receives the encrypted data transmitted by the first server; and then transmits the encrypted data to the second application program of the first terminal. The second application program transmits the message decryption request to the first server, and obtains the decrypted message data through the first server. In this way, during message sharing between the first application program and the second application program, the encrypted data is transmitted, so that security of message transmission between different application programs can be effectively ensured.

[00139] In some possible implementations, the message data is encrypted by the first server using the following method:

selecting an encryption method for the message data from a plurality of candidate encryption methods, and encrypting the message data using the selected encryption method, to obtain the encrypted data.

[00140] The encrypted data further includes identification information. The identification information indicates an identifier corresponding to the message data and an identifier corresponding to the encryption method.

[00141] Specifically, the first server may be configured to encrypt messages transmitted by a plurality of different application programs, and different messages may correspond to different

encryption methods. Therefore, different messages and encryption methods need to be marked with identification information. In this way, when the message decryption request is received, a corresponding decryption method can be determined for a corresponding encrypted message.

[00142] The obtaining decrypted message data through the first server based on the encrypted data includes:

transmitting a message decryption request to the first server, the message decryption request including the identification information; and receiving the decrypted data returned by the first server, the first server querying a plurality of candidate decryption methods for a decryption method corresponding to the identification information, and obtaining the decrypted data using the decryption method; and

obtaining the decrypted message data based on the decrypted data.

[00143] Specifically, a plurality of candidate encryption methods may be pre-stored in the first server. When the message encryption request is received, one encryption method may be randomly selected to encrypt the message data, and the identifier of the message data and the identifier of the selected encryption method are stored in association. When the message decryption request is received, the corresponding encryption method may be determined based on the identification information carried in the message decryption request, to determine the corresponding decryption method. The first server may decrypt the encrypted data using the decryption method, or may transmit the decryption method to the second client, and the second client decrypts the encrypted data using the decryption method.

[00144] In some possible implementations, the candidate encryption methods correspond to a plurality of security levels. The message encryption request further includes at least one of a first identifier of the first application program, a second identifier of the second application program, and security level selection information for the message data.

[00145] The selecting an encryption method for the message data from a plurality of candidate encryption methods includes:

selecting, based on at least one of a first application type of the first application program, a second application type corresponding to the second identifier, and the security level selection information for the message data, a security level from a plurality of security levels; and

determining an encryption method corresponding to the selected security level from the plurality of candidate encryption methods.

[00146] Specifically, correspondences among different sharer application programs (namely,

the first application programs), different sharee application programs (namely, the second application programs), and the security levels may be preset. Alternatively, the sharer provides the security level selection information, that is, the sharer may select a target security level, and the first server may determine the corresponding encryption method based on the determined target security level or the target security level selected by the sharer.

[00147] In some possible implementations, the message encryption request further includes the first identifier of the first application program and the second identifier of the second application program.

[00148] The first server encrypts the message data using the following method:

determining a first message format of the first application program that corresponds to the first identifier, and determining a second message format of the second application program that corresponds to the second identifier; and

directly encrypting the message data to obtain the encrypted data if the first message format is the same as the second message format.

[00149] Specifically, if the first message format is the same as the second message format, format conversion does not need to be performed on the message data, the message data may be directly encrypted, and the decrypted message data may be directly shared in the second application program.

[00150] In some possible implementations, that the first server encrypts the message data further includes:

if the first message format is different from the second message format, converting the message data into a target format corresponding to the second message format (that is, converting a data format of the message data according to the second message format); and

encrypting the message data in the target format to obtain the encrypted data.

[00151] Specifically, if the first message format is different from the second message format, the message data needs to be converted into the target format corresponding to the second message format of the second application type. In this way, the message data in the converted format can be displayed in the second application program and viewed by the sharee.

[00152] In some possible implementations, the message encryption request further includes the second identifier of the second application program.

[00153] The first server encrypts the message data using the following method:

determining the second message format of the second application type that corresponds to the second identifier;

converting the message data according to the second message format; and

encrypting the message data in the second message format, to obtain the encrypted data.

[00154] Specifically, alternatively, the message data may be directly converted into the target format corresponding to the second message format without determining whether the first message format of the first application program is the same as the second message format of the second application program. This can ensure that the sharee can normally view the message data through the second application program.

[00155] In some possible implementations, the first server encrypts the message data using the following method:

converting the message data into a preset message format, and encrypting the message data in the preset message format, to obtain the encrypted data.

[00156] The preset message format may be a universal format applicable to all application programs.

[00157] Specifically, alternatively, the message data may be directly converted into the universal preset message format without determining whether the first message format of the first application program is the same as the second message format of the second application program and without the need for determining the second message format of the second application program. This can omit the process of determining the first message format or the second message format while ensuring that the sharee can normally view the message data through the second application program, thereby improving message format conversion efficiency.

[00158] In some possible implementations, the message decryption request includes the encrypted data.

[00159] The encrypted data is decrypted using the following method:

decrypting the encrypted data through the first server in response to the message decryption request, to obtain the decrypted message data; and

returning the decrypted message data as the decrypted data to the second application program.

[00160] Specifically, the first server may receive the encrypted data transmitted by the

second application program. The first server directly decrypts the encrypted data to obtain the decrypted message data, and then returns the message data to the second application program.

[00161] In some possible implementations, the message decryption request includes the identification information. The identification information indicates an identifier corresponding to the message data and an identifier corresponding to the encryption method for the message data.

[00162] The encrypted data is decrypted using the following method:

querying, through the first server, for a decryption method corresponding to the identification information; and

transmitting the decryption method as the decrypted data to the second application program, to cause the second application program to decrypt the encrypted data using the decryption method, to obtain the decrypted message data, and share the message data to the sharee through the second server.

[00163] Specifically, alternatively, the first server may receive the identification information transmitted by the second application program. The first server queries for the decryption method corresponding to the identification information, and returns the decryption method to the second application program, and the second application program decrypts the encrypted data using the decryption method, to obtain the decrypted message data.

[00164] In some possible implementations, the encrypted data and the object identifier are transmitted to the second application program using the following method:

[00165] (1) Transmit a first verification request to the second application program.

[00166] The first verification request includes the first identifier of the first application program.

[00167] Specifically, the first application program starts the second application program through an operating system interface of the first terminal, and transmits the first verification request to the second application program.

[00168] (2) Receive a second verification request transmitted by the second application program.

[00169] The second verification request includes the second identifier of the second application program. The second verification request is transmitted to the first application program after the first application program passes first validity verification performed by the second application program based on the first identifier.

[00170] (3) Perform second validity verification on the second application program based on the second identifier, and transmit the encrypted data and the object identifier to the second application program if the second application program passes the second validity verification.

[00171] Specifically, before the first application program transmits the encrypted data to the second application program, the first application program and the second application program need to perform mutual verification.

[00172] In a specific implementation, the first application program first transmits the first verification request to the second application program, to transmit the first identifier of the first application program to the second application program. The second application program verifies validity of the first application program based on the first identifier. If the first application program passes the first validity verification, the second application program transmits the second identifier of the second application program to the first application program. Similarly, the first application program verifies validity of the second application program based on the second identifier. If the second application program also passes the second validity verification, secure communication may be established between the first application program and the second application program. That is, the two application programs successfully handshake. Then, the first application program transmits the encrypted data and the to-be-shared object identifier to the second application program through inter-process communication.

[00173] According to the message sharing method, when intending to share a message to the sharee of the second application program, the first application program of the first terminal transmits the message encryption request to the first server, encrypts the message data through the first server to obtain the encrypted data, and receives the encrypted data transmitted by the first server; and then transmits the encrypted data to the second application program of the first terminal. The second application program transmits the message decryption request to the first server, and obtains the decrypted message data through the first server. In this way, during message sharing between the first application program and the second application program, the encrypted data is transmitted, so that security of message transmission between different application programs can be effectively ensured.

[00174] Further, the first message format of the first application program is compared with the second message format of the second application program, and if the first message format is the same as the second message format, the message data is directly encrypted to obtain the encrypted data. If the first message format is different from the second message format, the message data is converted into the target format corresponding to the second message format, and then the message

data in the target format is encrypted to obtain the encrypted data, so that the sharee can normally view the message data through the second application program.

[00175] Further, alternatively, the message data may be directly converted into the target format corresponding to the second message format without determining whether the first message format of the first application program is the same as the second message format of the second application program. This can ensure that the sharee can normally view the message data through the second application program.

[00176] Further, alternatively, the message data may be directly converted into the universal preset message format without determining whether the first message format of the first application program is the same as the second message format of the second application program and without the need for determining the second message format of the second application program. This can omit the process of determining the first message format or the second message format while ensuring that the sharee can normally view the message data through the second application program, thereby improving message format conversion efficiency.

[00177] As shown in FIG. 8, in some possible implementations, a message sharing apparatus is provided, including:

a first receiving module 801, configured to receive a message encryption request transmitted by a first application program of a first terminal, the message encryption request including message data;

a first encryption module 802, configured to encrypt the message data to obtain encrypted data;

a first transmission module 803, configured to transmit the encrypted data to the first application program, to cause the first application program to transmit an object identifier of a sharee and the encrypted data to a second application program of the first terminal, the first application program being different from the second application program;

a first sharing module 804, configured to receive a message decryption request transmitted by the second application program of the first terminal; and return decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee corresponding to the object identifier.

[00178] In some possible implementations, when encrypting the message data to obtain the encrypted data, the first encryption module 802 is specifically configured to:

select an encryption method for the message data from a plurality of candidate encryption methods; and

encrypt the message data using the selected encryption method, to obtain the encrypted data, the encrypted data further including identification information; and the identification information indicating an identifier corresponding to the message data and an identifier corresponding to the encryption method.

[00179] The message decryption request includes the identification information. When returning decrypted data to the second application program in response to the message decryption request, the first sharing module 804 is specifically configured to:

query a plurality of candidate decryption methods for a decryption method corresponding to the identification information; and

return the decrypted data to the second application program based on the decryption method.

[00180] In some possible implementations, the candidate encryption methods correspond to a plurality of security levels. The message encryption request further includes at least one of a first identifier of the first application program, a second identifier of the second application program, and security level selection information for the message data.

[00181] When selecting the encryption method for the message data from the plurality of candidate encryption methods, the first encryption module 802 is specifically configured to:

select, based on at least one of a first application type of the first application program, a second application type corresponding to the second identifier, and the security level selection information for the message data, a security level from the plurality of security levels; and

determine an encryption method corresponding to the selected security level from the candidate encryption methods.

[00182] In some possible implementations, the message encryption request further includes the first identifier of the first application program and the second identifier of the second application program.

[00183] When encrypting the message data to obtain the encrypted data, the first encryption module 802 is specifically configured to:

determine a first message format corresponding to the first identifier, and determine a second message format corresponding to the second identifier; and

directly encrypt the message data to obtain the encrypted data if the first message format is the same as the second message format.

[00184] In some possible implementations, when encrypting the message data to obtain the encrypted data, the first encryption module 802 is further configured to:

convert a data format of the message data according to the second message format; and

encrypt the message data in the converted data format, to obtain the encrypted data.

[00185] In some possible implementations, the message encryption request further includes the second identifier of the second application program.

[00186] When encrypting the message data to obtain the encrypted data, the first encryption module 802 is specifically configured to:

determine the second message format corresponding to the second identifier;

convert the data format of the message data according to the second message format; and

encrypt the message data in the converted data format, to obtain the encrypted data.

[00187] In some possible implementations, when encrypting the message data to obtain the encrypted data, the first encryption module 802 is specifically configured to:

convert the message format of the message data into a preset message format; and encrypt the message data in the preset message format, to obtain the encrypted data.

[00188] In some possible implementations, the message decryption request includes the encrypted data.

[00189] When returning the decrypted data to the second application program in response to the message decryption request, the first sharing module 804 is specifically configured to:

decrypting the encrypted data in response to the message decryption request, to obtain the decrypted message data; and

returning the decrypted message data as the decrypted data to the second application program.

[00190] In some possible implementations, the message decryption request includes the identification information. The identification information indicates the identifier corresponding to the message data and the identifier corresponding to the encryption method for the message data.

[00191] When returning the decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through the second server based on the decrypted data, the decrypted message data to the sharee, the first sharing module 804 is specifically configured to:

query for a decryption method corresponding to the identification information; and

transmit the decryption method as the decrypted data to the second application program, to cause the second application program to decrypt the encrypted data using the decryption method, to obtain the decrypted message data, and share the message data to the sharee through the second server.

[00192] In some possible implementations, the encrypted data and the object identifier are transmitted by the first application program to the second application program using the following method:

transmitting the first identifier of the first application program to the second application program;

receiving a verification request transmitted by the second application program, the verification request including the second identifier of the second application program; and the verification request being transmitted to the first application program after the first application program passes first validity verification performed by the second application program based on the first identifier; and

performing second validity verification on the second application program based on the second identifier, and transmitting the encrypted data to the second application program if the second application program passes the second validity verification.

[00193] According to the message sharing apparatus, when intends to share a message to the sharee of the second application program, the first application program of the first terminal transmits the message encryption request to the first server, encrypts the message data through the first server to obtain the encrypted data, and receives the encrypted data transmitted by the first server; and then transmits the encrypted data to the second application program of the first terminal. The second application program transmits the message decryption request to the first server, and obtains the decrypted message data through the first server. In this way, during message sharing between the first application program and the second application program, the encrypted data is transmitted, so that security of message transmission between different application programs can be effectively ensured.

[00194] Further, the first message format of the first application program is compared with the second message format of the second application program, and if the first message format is the same as the second message format, the message data is directly encrypted to obtain the encrypted data. If the first message format is different from the second message format, the message data is converted into the target format corresponding to the second message format, and then the message data in the target format is encrypted to obtain the encrypted data, so that the sharee can normally view the message data through the second application program.

[00195] Further, alternatively, the message data may be directly converted into the target format corresponding to the second message format without determining whether the first message format of the first application program is the same as the second message format of the second application program. This can ensure that the sharee can normally view the message data through the second application program.

[00196] Further, alternatively, the message data may be directly converted into the universal preset message format without determining whether the first message format of the first application program is the same as the second message format of the second application program and without the need for determining the second message format of the second application program. This can omit the process of determining the first message format or the second message format while ensuring that the sharee can normally view the message data through the second application program, thereby improving message format conversion efficiency.

[00197] As shown in FIG. 9, in some possible implementations, a message sharing apparatus is provided, including:

a second transmitting module 901, configured to transmit a message encryption request to a first server in response to a message sharing operation, the message encryption request including message data;

a second receiving module 902, configured to receive encrypted data that is returned by the first server in response to the message encryption request, the encrypted data being obtained by the first server encrypting the message data; and

a second sharing module 903, configured to transmit the encrypted data and an object identifier of a sharee to a second application program, to cause the second application program to obtain decrypted message data through the first server based on the encrypted data, and share, through the second server, the message data to the sharee corresponding to the object identifier, the first application program being different from the second application program.

[00198] In some possible implementations, the message data is encrypted by the first server using the following method:

determining an encryption method for the message data from a plurality of candidate encryption methods, and encrypting the message data using the encryption method, to obtain the encrypted data, the encrypted data further including identification information; and the dentification information indicating an identifier corresponding to the message data and an identifier corresponding to the encryption method.

[00199] The obtaining decrypted message data through the first server based on the encrypted data includes:

transmitting a message decryption request to the first server, the message decryption request including the identification information; and receiving the decrypted data returned by the first server, the first server querying a plurality of candidate decryption methods for a decryption method corresponding to the identification information, and obtaining the decrypted data using the decryption method; and

obtaining the decrypted message data based on the decrypted data.

[00200] In some possible implementations, the candidate encryption methods correspond to a plurality of security levels. The message encryption request further includes at least one of a first identifier of the first application program, a second identifier of the second application program, and security level selection information for the message data.

[00201] The determining an encryption method for the message data from a plurality of candidate encryption methods includes:

determining, based on at least one of a first application type of the first application program, a second application type corresponding to the second identifier, and the security level selection information for the message data, a corresponding target security level; and

determining the encryption method corresponding to the target security level from the candidate encryption methods.

[00202] In some possible implementations, the message encryption request further includes the first identifier of the first application program and the second identifier of the second application program.

[00203] The first server encrypts the message data using the following method:

determining a first message format corresponding to the first identifier, and

determining a second message format corresponding to the second identifier; and

directly encrypting the message data to obtain the encrypted data if the first message format is the same as the second message format.

[00204] In some possible implementations, that the first server encrypts the message data further includes:

if the first application type is different from the second application type, converting the message data into a target format corresponding to the second application type; and

encrypting the message data in the target format to obtain encrypted data.

[00205] In some possible implementations, the message encryption request further includes the second identifier of the second application program.

[00206] The first server encrypts the message data using the following method:

determining the second message format corresponding to the second identifier;

converting the message data into the target format corresponding to the second application type; and

encrypting the message data in the target format to obtain the encrypted data.

[00207] In some possible implementations, the first server encrypts the message data using the following method:

converting the message data into a preset message format, and encrypting the message data in the preset message format, to obtain the encrypted data.

[00208] In some possible implementations, the message decryption request includes the encrypted data.

[00209] The encrypted data is decrypted using the following method:

decrypting the encrypted data through the first server in response to the message decryption request, to obtain the decrypted message data; and

returning the decrypted message data as the decrypted data to the second application program.

[00210] In some possible implementations, the message decryption request includes the identification information. The identification information indicates the identifier corresponding to the message data and the identifier corresponding to the encryption method for the message data.

[00211] The encrypted data is decrypted using the following method:

querying, through the first server, for a decryption method corresponding to the first identifier; and

transmitting the decryption method as the decrypted data to the second application program, to cause the second application program to decrypt the encrypted data using the decryption method, to obtain the decrypted message data, and share the message data to the sharee through the second server.

[00212] In some possible implementations, the encrypted data and the object identifier are transmitted to the second application program using the following method:

transmitting the first identifier of the first application program to the second application program;

receiving a verification request transmitted by the second application program, the verification request including the second identifier of the second application program; and the verification request being transmitted to the first application program after the first application program passes first validity verification performed by the second application program based on the first identifier; and

performing second validity verification on the second application program based on the second identifier, and transmitting the encrypted data to the second application program if the second application program passes the second validity verification.

[00213] According to the message sharing apparatus, when intends to share a message to the sharee of the second application program, the first application program of the first terminal transmits the message encryption request to the first server, encrypts the message data through the first server to obtain the encrypted data, and receives the encrypted data transmitted by the first server; and then transmits the encrypted data to the second application program of the first terminal. The second application program transmits the message decryption request to the first server, and obtains the decrypted message data through the first server. In this way, during message sharing between the first application program and the second application program, the encrypted data is transmitted, so that security of message transmission between different application programs can be effectively ensured.

[00214] Further, the first message format of the first application program is compared with the second message format of the second application program, and if the first message format is the same as the second message format, the message data is directly encrypted to obtain the encrypted

data. If the first message format is different from the second message format, the message data is converted into the target format corresponding to the second message format, and then the message data in the target format is encrypted to obtain the encrypted data, so that the sharee can normally view the message data through the second application program.

[00215] Further, alternatively, the message data may be directly converted into the target format corresponding to the second message format without determining whether the first message format of the first application program is the same as the second message format of the second application program. This can ensure that the sharee can normally view the message data through the second application program.

[00216] Further, alternatively, the message data may be directly converted into the universal preset message format without determining whether the first message format of the first application program is the same as the second message format of the second application program and without the need for determining the second message format of the second application program. This can omit the process of determining the first message format or the second message format while ensuring that the sharee can normally view the message data through the second application program, thereby improving message format conversion efficiency.

[00217] The apparatus provided in the embodiments of this application may perform the method provided in the embodiments of this application, and an implementation principle of the apparatus is similar to that of the method. Actions performed by the modules in the apparatus provided in the embodiments of this application correspond to the operations in the method provided in the embodiments of this application. For detailed descriptions of functions of the modules of the apparatus, refer to the foregoing description of the corresponding method. Details are not described herein again.

[00218] The embodiments of this application provide an electronic device, including a memory, a processor, and a computer program that is stored in the memory. The processor may execute the computer program stored in the memory to implement the method according to any one of the foregoing exemplary embodiments.

[00219] An exemplary embodiment provides an electronic device. As shown in FIG. 10, an electronic device 4000 shown in FIG. 10 includes: a processor 4001 and a memory 4003. The processor 4001 is connected to the memory 4003, for example, through a bus 4002. In an embodiment, the electronic device 4000 may further include a transceiver 4004. The transceiver 4004 may be configured to implement data interaction between the electronic device and another electronic device, such as data transmission and/or data receiving. In practical application, one or

more transceivers 4004 may be provided, and a structure of the electronic device 4000 does not constitute a limitation on the embodiments of this application.

[00220] The processor 4001 may be a central processing unit (CPU), a general-purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or another programmable logic device, a transistor logic device, a hardware component, or any combination thereof. The processor may implement or execute various exemplary logical blocks, modules, and circuits described with reference to content disclosed in this application. The processor 4001 may alternatively be a combination that implements a computing function, such as a combination of one or more microprocessors, or a combination of a DSP and a microprocessor.

[00221] The bus 4002 may include a path for transmitting information between the foregoing components. The bus 4002 may be a peripheral component interconnect (PCI) bus, an extended industry standard architecture (EISA) bus, or the like. The bus 4002 may be classified into an address bus, a data bus, a control bus, or the like. For ease of representation, only the bus is indicated by only one thick line in FIG. 10, but it does not indicate that only one bus or one type of bus exists.

[00222] The memory 4003 may be a read only memory (ROM) or another type of static storage device capable of storing static information and instructions, a random access memory (RAM) or another type of dynamic storage device capable of storing information and instructions, an electrically erasable programmable read-only memory (EEPROM), a compact disc read only memory (CD-ROM) or another optical disk storage device, an optical disc storage device (including a compact disc, a laser disc, an optical disc, a digital versatile disc, a Blu-ray disc, and the like), a magnetic disk storage medium or another magnetic storage device, or any other medium that is capable of carrying or storing a computer program and that is accessible by a computer. This is not limited herein.

[00223] The memory 4003 is configured to store the computer program for implementing the embodiments of this application, which is controlled and executed by the processor 4001. The processor 4001 is configured to execute the computer program stored in the memory 4003, to implement the operations shown in the foregoing method embodiments.

[00224] Based on the principle the same as that of the method provided in the embodiments of this application, the embodiments of this application provide a computer-readable storage medium. The computer-readable storage medium has a computer program stored therein, and the computer program, when executed by a processor, may implement corresponding content of the

foregoing method embodiments.

[00225] The embodiments of this application further provide a computer program product. The computer program product includes a computer program. The computer program, when executed by a processor, may implement corresponding content of the foregoing method embodiments.

[00226] In the description, claims, and accompanying drawings of this application, the terms such as "first", "second", "third", "fourth", "1", "2", and the like (if any) are intended to distinguish between similar objects but do not necessarily indicate a specific order or sequence. Data used in this way is exchangeable in a proper case, so that the embodiments of this application described herein can be implemented in an order other than those illustrated or described in the drawings or text.

[00227] Although the operations are indicated by the arrows in the flowcharts of the embodiments of this application, the implementation order of these operations is not limited to the order indicated by the arrows. Unless explicitly specified, in some implementation scenarios of the embodiments of this application, the implementation operations in the flowcharts may be performed in another order as required. In addition, some or all of the operations in the flowcharts may include a plurality of sub-operations or a plurality of stages based on a practical implementation scenario. Some or all of these sub-operations or stages may be performed at the same moment, and each of these sub-operations or stages may be separately performed at a different moment. In a scenario with different implementation moments, an implementation order of these sub-operations or stages may be flexibly configured as required. This is not limited in the embodiments of this application.

[00228] The above are merely exemplary implementations of some implementation scenarios of this application. A person of ordinary skill in the art may make other similar implementations based on the technical ideas of this application without departing from the concepts of the technical solutions of this application, and these implementations still fall within the scope of protection of the embodiments of this application.

CLAIMS

What is claimed is:

1. A message sharing method, applied to a first server and comprising:

receiving a message encryption request transmitted by a first application program of a first terminal, the message encryption request comprising message data;

encrypting the message data to obtain encrypted data;

transmitting the encrypted data to the first application program, to cause the first application program to transmit an object identifier of a sharee and the encrypted data to a second application program of the first terminal, the first application program being different from the second application program;

receiving a message decryption request transmitted by the second application program of the first terminal; and

returning decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee corresponding to the object identifier.

2. The method according to claim 1, wherein the encrypting the message data to obtain encrypted data comprises:

selecting an encryption method for the message data from a plurality of candidate encryption methods; and

encrypting the message data using the selected encryption method, to obtain the encrypted data, the encrypted data further comprising identification information; and the identification information indicating an identifier corresponding to the message data and an identifier corresponding to the encryption method;

the message decryption request comprises the identification information; and the returning decrypted data to the second application program in response to the message decryption request comprises:

querying a plurality of candidate decryption methods for a decryption method corresponding to the identification information; and

returning the decrypted data to the second application program based on the decryption method.

3. The method according to claim 2, wherein the candidate encryption methods correspond to a plurality of security levels; the message encryption request further comprises at least one of a first identifier of the first application program, a second identifier of the second application program, and security level selection information for the message data; and

the selecting an encryption method for the message data from a plurality of candidate encryption methods comprises:

selecting, based on at least one of a first application type of the first application program, a second application type corresponding to the second identifier, and the security level selection information for the message data, a security level from the plurality of security levels; and

determining an encryption method corresponding to the selected security level from the plurality of candidate encryption methods.

4. The method according to any one of claims 1 to 3, wherein the message encryption request further comprises the first identifier of the first application program and the second identifier of the second application program; and

the encrypting the message data to obtain encrypted data comprises:

determining a first message format of the first application program that corresponds to the first identifier, and determining a second message format of the second application program that corresponds to the second identifier; and

directly encrypting the message data to obtain the encrypted data if the first message format is the same as the second message format.

5. The method according to claim 4, wherein the encrypting the message data to obtain encrypted data further comprises:

converting a data format of the message data according to the second message format if the first message format is different from the second message format; and

encrypting the message data in the converted data format, to obtain the encrypted data.

6. The method according to any one of claims 1 to 5, wherein the message encryption request further comprises the second identifier of the second application program; and

the encrypting the message data to obtain encrypted data comprises:

determining the second message format of the second application type that corresponds to the second identifier;

converting the data format of the message data according to the second message format; and

encrypting the message data in the converted data format, to obtain the encrypted data.

7. The method according to any one of claims 1 to 6, wherein the encrypting the message data to obtain encrypted data comprises:

converting the data format of the message data into a preset message format; and

encrypting the message data in the preset message format, to obtain the encrypted data.

8. The method according to any one of claims 1 to 7, wherein the message decryption request comprises the encrypted data; and

the returning decrypted data to the second application program in response to the message decryption request comprises:

decrypting the encrypted data in response to the message decryption request, to obtain the decrypted message data; and

returning the decrypted message data as the decrypted data to the second application program.

9. The method according to any one of claims 1 to 8, wherein the message decryption request comprises the identification information; and the identification information indicates the identifier corresponding to the message data and the identifier corresponding to the encryption method for the message data; and

the returning decrypted data to the second application program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee comprises:

querying for the decryption method corresponding to the identification information; and

transmitting the decryption method as the decrypted data to the second application program, to cause the second application program to decrypt the encrypted data using the decryption method, to obtain the decrypted message data, and share the message data to the sharee through the second server.

10. The method according to any one of claims 1 to 9, wherein the encrypted data and the object identifier are transmitted to the second application program by the first application program using the following method:

transmitting a first verification request to the second application program, the first verification request comprising the first identifier of the first application program;

receiving a second verification request transmitted by the second application program, the second verification request comprising the second identifier of the second application program; and the second verification request being transmitted to the first application program after the first application program passes first validity verification performed by the second application program based on the first identifier; and

performing second validity verification on the second application program based on the second identifier, and transmitting the encrypted data and the object identifier to the second application program if the second application program passes the second validity verification.

11. A message sharing method, applied to a first application program of a first terminal and comprising:

transmitting a message encryption request to a first server in response to a message sharing operation, the message encryption request comprising message data;

receiving encrypted data that is returned by the first server in response to the message encryption request, the encrypted data being obtained by the first server encrypting the message data; and

transmitting the encrypted data and an object identifier of a sharee to a second application program, to cause the second application program to obtain decrypted message data through the first server based on the encrypted data, and share, through a second server, the message data to the sharee corresponding to the object identifier, the first application program being different from the second application program.

12. A message sharing apparatus, comprising:

a first receiving module, configured to receive a message encryption request transmitted by a first application program of a first terminal, the message encryption request comprising message data;

a first encryption module, configured to encrypt the message data to obtain encrypted data;

a first transmission module, configured to transmit the encrypted data to the first application program, to cause the first application program to transmit an object identifier of a sharee and the encrypted data to a second application program of the first terminal, the first application program being different from the second application program; and

a first sharing module, configured to receive a message decryption request transmitted by a second application program of the first terminal; and return decrypted data to the second application

program in response to the message decryption request, to cause the second application program to share, through a second server based on the decrypted data, decrypted message data to the sharee corresponding to the object identifier.

13. A message sharing apparatus, comprising:

a second transmission module, configured to transmit a message encryption request to a first server in response to a message sharing operation, the message encryption request comprising message data;

a second receiving module, configured to receive encrypted data that is returned by the first server in response to the message encryption request, the encrypted data being obtained by the first server encrypting the message data; and

a second sharing module, configured to transmit the encrypted data and an object identifier of a sharee to a second application program, to cause the second application program to obtain decrypted message data through the first server based on the encrypted data, and share, through a second server, the message data to the sharee corresponding to the object identifier, the first application program being different from the second application program.

14. An electronic device, comprising a memory and a processor, the memory having a computer program stored therein, and the processor, when executing the computer program, implementing the method according to any one of claims 1 to 11.

15. A computer-readable storage medium, having a computer program stored therein, the computer program, when executed by a processor, implementing the method according to any one of claims 1 to 11.

16. A computer program product, comprising a computer program, the computer program, when executed by a processor, implementing operations of the method according to any one of claims 1 to 11.