

D E S C R I P T I O N

RE-ENCRYPTION KEY GENERATOR, RE-ENCRYPTION DEVICE,
ENCRYPTION DEVICE, DECRYPTION DEVICE, AND PROGRAM

5

Technical Field

Embodiments described herein relate generally to a re-encryption key generator, a re-encryption device, an encryption device, a decryption device, and a program.

10

Background Art

In a file sharing system in which a file is shared among a plurality of users after a user uploads the file to a server, a method of re-encryption is known to maintain secrecy of file with respect to the server.

15

Citation List

Non Patent Literature

Non Patent Literature 1: R. Hayashi, T. Matsushita, T. Yoshida, Y. Fujii, and K. Okada, "Unforgeability of Re-Encryption Keys against Collusion Attack in Proxy Re-Encryption," In Proc. IWSEC 2011, LNCS 7038, pp. 210-229, Springer-Verlag, 2011.

20

Summary of Invention

Technical Problem

A conventional re-encryption scheme presents no particular problem, but if a problem should arise, a server is enabled to create re-encrypted text without the permission of each user. Therefore, from the viewpoint of

25

realizing safe file sharing, there is room for improvements in a proxy re-encryption scheme.

A re-encryption key generator according to an embodiment generates a re-encryption key to obtain re-encrypted data that can be decrypted by a second private
5 key of a second user device by re-encrypting ciphertext obtained by encrypting plaintext by a first public key of a first user device without decryption.

The re-encryption key generator includes a first
10 storage means, a second storage means, and a re-encryption key generation means.

The first storage means stores a first private key corresponding to the first public key.

The second storage means stores a second re-encryption
15 key generation key of the second user device that is different from a second public key corresponding to the second private key.

The re-encryption key generation means generates the re-encryption key based on the first private key and the
20 second re-encryption key generation key.

Brief Description of Drawings

FIG. 1 is a schematic diagram showing a configuration of a re-encryption system according to a first embodiment.

FIG. 2 is a sequence diagram illustrating an operation
25 of key setup processing in the embodiment.

FIG. 3 is a sequence diagram illustrating the operation of encryption processing in the embodiment.

FIG. 4 is a sequence diagram illustrating the operation of re-encryption key generation processing in the embodiment.

5 FIG. 5 is a sequence diagram illustrating the operation of re-encryption processing in the embodiment.

FIG. 6 is a sequence diagram illustrating the operation of decryption processing in the embodiment.

FIG. 7 is a diagram illustrating the configuration of a file sharing system according to the embodiment.

10 FIG. 8 is a schematic diagram illustrating the file sharing system in the embodiment.

FIG. 9 is a diagram illustrating the configuration of a file sharing system according to the embodiment.

15 FIG. 10 is a diagram illustrating the configuration of a file sharing system according to the embodiment.

FIG. 11 is a schematic diagram showing the configuration of a re-encryption key generator according to the embodiment.

20 FIG. 12 is a schematic diagram showing the configuration of a re-encryption system according to a thirteenth embodiment.

FIG. 13 is a sequence diagram illustrating the operation of key setup processing in the embodiment.

25 FIG. 14 is a sequence diagram illustrating the operation of encryption processing in the embodiment.

FIG. 15 is a sequence diagram illustrating the operation of re-encryption key generation processing in the

embodiment.

FIG. 16 is a sequence diagram illustrating the operation of re-encryption processing in the embodiment.

FIG. 17 is a sequence diagram illustrating the operation of decryption processing in the embodiment.

FIG. 18 is a schematic diagram illustrating a file sharing system in the embodiment.

Description of Embodiments

Hereinafter, each embodiment will be described using the drawings. Each device shown below can be carried out by either of a hardware configuration and a combined configuration of hardware resources and software. As software of the combined configuration, as shown in FIG. 1 or 12, a program causing each computer to realize the function of each device by being installed on each computer from a network or one of non-transitory computer-readable storage media M1 to M5 in advance and executed by the processor of each computer is used.

(First embodiment)

First, a proxy re-encryption scheme will be described. The basic model of the proxy re-encryption scheme includes five functions (hereinafter, also called algorithms) of key generation, encryption, decryption, re-encryption key generation, and re-encryption, shown below. The functions of key generation, encryption, and decryption are the same as functions of normal public key cryptography.

(Key generation) $\text{KeyGen}(\lambda) \rightarrow (\text{pk}, \text{sk})$

The key generation algorithm KeyGen outputs a set (pk, sk) of a public key pk and a private key sk when a security parameter λ is input.

(Encryption) $Enc(pk_A, m) \rightarrow C_A$

5 The encryption algorithm Enc outputs ciphertext C_A addressed to a user A when a public key pk_A of a user A and a message m are input.

(Decryption) $Dec(sk_A, C_A) \rightarrow m$

10 The decryption algorithm Dec outputs the message m when a private key sk_A of the user A and the ciphertext C_A addressed to the user A are input.

(Re-encryption key generation)

$ReKeyGen(pk_A, sk_A, pk_B, sk_B) \rightarrow rk_{A \rightarrow B}$

15 The re-encryption key generation algorithm ReKeyGen outputs a re-encryption key $rk_{A \rightarrow B}$ when the public key pk_A of the user A, the private key sk_A of the user A, a public key pk_B of a user B, and a private key sk_B of the user B are input.

(Re-encryption) $ReEnc(rk_{A \rightarrow B}, C_A) \rightarrow C_B$

20 The re-encryption algorithm ReEnc outputs ciphertext C_B addressed to the user B when the re-encryption key $rk_{A \rightarrow B}$ and the ciphertext C_A addressed to the user A are input.

25 The above is the basic model. However, depending on the implementation method of re-encryption, a model that is different in input into the functions or a model including other functions or keys than the above can also be considered.

For example, like the re-encryption scheme shown in the present embodiment, public parameters (details will be described later) may be included for input into each function of the encryption, decryption, re-encryption key generation, and re-encryption.

For example, like the re-encryption scheme shown in the present embodiment, a non-interactive model that makes the private key sk_B of the user B unnecessary for input of the re-encryption key generation algorithm $ReKeyGen$ is considered. Also, a model into which, instead of the private key sk_A of the user A, the re-encryption key $rk_{A \rightarrow B}$ addressed to the user B and a private key sk_C of a user C are input is also considered.

In addition, a unidirectional model in which while re-encryption of the ciphertext $C_A \rightarrow C_B$ can be performed by the re-encryption key $rk_{A \rightarrow B}$, the inverse conversion of the ciphertext $C_B \rightarrow C_A$ cannot be performed, and a bidirectional model, in which the inverse conversion can also be performed, are known. In the bidirectional model, the re-encryption key $rk_{A \rightarrow B}$ may also be represented as $rk_{A \leftrightarrow B}$.

Further, in public key cryptography, a scheme based on ID-based encryption is also considered. In this case, function setup for master key generation increases and the master key and ID are added to the input of the key generation algorithm $KeyGen$. In the ID-based encryption, the public key pk is the ID itself.

In the re-encryption key generation algorithm $ReKeyGen$,

like the re-encryption scheme shown in the present embodiment, instead of the public key pk_B of the user B, the re-encryption key generation key rk_B of the user B may be input. In this case, the key generation algorithm
5 KeyGen and the re-encryption key generation algorithm ReKeyGen are represented as shown below.

(Key generation) $\text{KeyGen}(\lambda) \rightarrow (pk, sk, rk, par)$

The key generation algorithm KeyGen outputs a set (pk, sk, rk) of the public key pk , the private key sk , and the
10 re-encryption key generation key rk_B and a public parameter par when the security parameter λ is input. In the present embodiment, this processing is called key setup processing. This also applies to embodiments and modifications described later.

15 (Re-encryption key generation)

$\text{ReKeyGen}(sk_A, rk_B, par) \rightarrow rk_{A \rightarrow B}$

The re-encryption key generation algorithm ReKeyGen outputs the re-encryption key $rk_{A \rightarrow B}$ when the private key sk_A of the user A, the re-encryption key generation key rk_B of
20 the user B, and the public parameter par are input.

Like the re-encryption scheme shown in the present embodiment, the public key pk_A of the user A and the public parameter par may be added as the input of the re-encryption algorithm ReEnc. In this case, the re-
25 encryption algorithm ReEnc is represented as shown below.

(Re-encryption) $\text{ReEnc}(rk_{A \rightarrow B}, C_A, pk_A, par) \rightarrow C_B$

The re-encryption algorithm ReEnc outputs the

ciphertext C_B addressed to the user B when the re-encryption key $rk_{A \rightarrow B}$, the ciphertext C_A addressed to the user A, the public key pk_A of the user A, and the public parameter par are input.

5 Next, The configuration of a file sharing system according to the present embodiment will be described using FIG. 7. The file sharing system includes a server S_v as an information processing apparatus owned by a service provider who provides a service and user apparatuses A, B, 10 C, ... (hereinafter, may be abbreviated as users A, B, C, ... or may also be called a first user apparatus, a second user apparatus, ...) as information processing apparatuses owned by users as clients of the service. The server S_v and the user apparatuses A, B. C... are connected via a 15 communication network. The communication network is, for example, a wireless LAN (Local Area Network), a cable LAN, an optical network, a telephone network, an intra-net, an Ethernet (registered trademark), the Internet, or a combination of the above networks. One unit of the server 20 S_v may be provided or a plurality of units thereof may be provided.

FIG. 8 is a schematic diagram illustrating the file sharing system. The file sharing system includes the server S_v , the user A, the user B, the user C, a group 25 administrator, and a user administrator. The user administrator is an entity that administers all users. The user administrator issues an ID and an initial password to

enable a user to log in to the file sharing system.

As a preparation, the group administrator is assumed to have a private key sk_{Gr} . Also, a public key pk_{Gr} of the group administrator is assumed to be stored in a public key storage unit 1 of the server Sv . Also, each user i is assumed to have a private key sk_i and a re-encryption key generation key rk_i . That is, the user A has sk_A and rk_A , the user B has sk_B and rk_B , and the user C has sk_C and rk_C . The server Sv and each user i can acquire the public parameter par and a public key pk_i of each user i , but the description thereof is omitted here and a detailed description will be provided later. In addition, file of plaintext to be shared is denoted as m .

The user C encrypts the file m by the public key pk_{Gr} of the group administrator stored in the public key storage unit 1 and uploads obtained ciphertext $E(pk_{Gr}, m)$ to the server Sv . Here, $E(PK, M)$ represents ciphertext obtained by encrypting plaintext M by using the public key PK . The server Sv stores the ciphertext in a ciphertext storage unit 2.

Now, assume that the user A approves sharing of the file m . The user A transmits the re-encryption key generation key rk_A of the user A to the group administrator. The group administrator generates a re-encryption key $rk_{Gr \rightarrow A}$ that re-encrypts ciphertext $E(pk_{Gr}, m)$ addressed to the group administrator into ciphertext $E(pk_A, m)$ addressed to the user A using the private key sk_{Gr} of the group

administrator and the re-encryption key generation key rk_A of the user A. The group administrator transmits the generated re-encryption key $rk_{Gr \rightarrow A}$ to the server Sv. The server Sv stores the re-encryption key $rk_{Gr \rightarrow A}$ in a re-
5 re-encryption key storage unit 3.

The user A transmits a request of re-encryption of the ciphertext $E(pk_{Gr}, m)$ to the server Sv. In response to the request received from the user A, the server Sv re-encrypts the ciphertext $E(pk_{Gr}, m)$ into re-encrypted text $E(pk_A, m)$
10 using the re-encryption key $rk_{Gr \rightarrow A}$ stored in the re-encryption key storage unit 3 and addressed to the user A and transmits the obtained re-encrypted text $E(pk_A, m)$ to the user A. The re-encrypted text is ciphertext obtained by re-encrypting ciphertext.

15 The user A decrypts the re-encrypted text $E(pk_A, m)$ downloaded from the server Sv by the private key sk_A held by the user A and uses the obtained file m.

The file sharing system as described above can hinder leakage of information in the process of file sharing
20 because the file m is not decrypted once between the encryption by the user C and the decryption by the user A.

In addition, there is no need for the user C to decide with which user to share a file when uploading ciphertext. That is, the user C only needs to encrypt the file m by the
25 public key pk_{Gr} of the group administrator and the key to be managed for encryption is only the public key pk_{Gr} of the group administrator and therefore, costs for key

management can be reduced.

In the present file sharing system, the server S_v does not have a decryption key sk_{Gr} to decrypt the ciphertext $E(pk_{Gr}, m)$. Therefore, the server S_v cannot decrypt the
5 ciphertext. Accordingly, threats of leakage by an invalid server S_v by decrypting ciphertext can be excluded and therefore, strict control of the server S_v can be eliminated, leading to a reduction in management costs.

In the proxy re-encryption scheme, the decryption
10 right of ciphertext is transferred to a transfer destination by re-encrypting ciphertext addressed to a transfer source (group administrator in the above example) into ciphertext addressed to the transfer destination (user A in the above example). In the proxy re-encryption scheme,
15 the public key of the transfer destination is included in the input of the re-encryption key generation algorithm $ReKeyGen$ (without the private key of the transfer destination being included). Thus, to enable, for example, the user i to generate ciphertext $E(pk_A, m')$ addressed to
20 the user A by encrypting other file m' using the public key pk_A of the user A, the public key pk_A of the user A is published in advance (for example, when the system is started up).

On the other hand, even if the user A does not approve
25 the transfer of the decryption right of ciphertext to the user A, the public key pk_A of the user A is already published and thus, the group administrator can generate

the re-encryption key $rk_{Gr \rightarrow A}$. Therefore, there is the possibility of re-encryption (that is, the transfer of the decryption right of ciphertext to the user A) without the approval of the user A.

5 In the present file sharing system, by contrast, the group administrator cannot generate the re-encryption key $rk_{Gr \rightarrow A}$ without the re-encryption key generation key rk_A being transmitted to the group administrator by the user A as the transfer destination. Therefore, re-encryption
10 (that is, the transfer of the decryption right of ciphertext to the transfer destination) can be prevented when no approval of the transfer destination is received.

The above also applies when the user B shares a file.

FIG. 1 is a schematic diagram showing the
15 configuration of a re-encryption system according to the first embodiment (the correspondence between FIG. 1 and FIG. 8 will be described later). The re-encryption system includes a key generator 10, an encryption device 20, a re-encryption key generator 30, a re-encryption device 40, and
20 a decryption device 50. The key generator 10 generates a public parameter of the re-encryption system and a pair of public key/private key and a re-encryption key generation key for each of the devices 30, 50 based on a security
25 parameter. Here, the re-encryption key generation key is a key that is different from the public key corresponding to the private key of each of the devices 30, 50. The term "re-encryption key generation key" may be changed to a

desired name such as "re-encryption key generation data" or "re-encryption key generation permission data".

5 The encryption device 20 encrypts plaintext data using the public key corresponding to the private key of the re-encryption key generator 30 and the public parameter and transmits the obtained ciphertext data to the re-encryption device 40.

10 The re-encryption key generator 30 generates a re-encryption key based on the private key of the re-encryption key generator 30 and the re-encryption key generation key of the decryption device 50 and transmits the obtained re-encryption key to the re-encryption device 40. The re-encryption key generation key of the decryption device 50 is different from the public key corresponding to the private key of the decryption device 50. In the present embodiment, the re-encryption key is generated based on the private key of the re-encryption key generator 30 and the re-encryption key generation key of the decryption device 50, the public parameter, and a random number. However, the random number is not required for generating a re-encryption key and may be omitted. For example, in a fifth or twelfth embodiment, a re-encryption key is generated without using any random number.

25 The re-encryption device 40 performs verification processing of ciphertext data received from the encryption device 20 based on the public key of the re-encryption key generator 30, the public parameter, and the re-encryption

key received from the re-encryption key generator 30 and also re-encrypts the ciphertext data received from the encryption device 20 without decryption and transmits the obtained re-encrypted data to the decryption device 50.

5 However, the verification processing is not required and may be omitted.

The decryption device 50 performs verification processing of re-encrypted data received from the re-encryption device 40 based on the private key of the local
10 device 50 and the public parameter and also obtains plaintext data by performing decryption processing of the re-encrypted data received from the re-encryption device 40. However, the verification processing is not required and may be omitted. Here, a case when one unit is provided for
15 each of the key generator 10, the encryption device 20, the re-encryption key generator 30, the re-encryption device 40, and the decryption device 50 is taken as an example, though a plurality of units may be provided for each. The encryption device 20 may encrypt plaintext data using the
20 public key of the encryption device 20 and the public parameter to transmit the obtained ciphertext data to the re-encryption device 40. In this case, the key generator 10 generates a public parameter, and a pair of public key/private key and a re-encryption key generation key for
25 each of the devices 20, 50. The re-encryption key generator 30 generates a re-encryption key based on the private key of the encryption device 20, the re-encryption

key generation key of the decryption device 50, the public parameter, and a random number and transmits the obtained re-encryption key to the re-encryption device 40.

5 The correspondence between FIG. 1 and FIG. 8 will be described. Each of the devices 10 to 50 is held by an entity that performs the relevant processing. When the user generates a set of a public key and private key and a re-encryption key generation key of the user, the relevant user holds the key generator 10. If, for example, the user administrator or the group administrator generates a set of
10 a public key and private key and a re-encryption key generation key of a user, the user administrator or the group administrator holds the key generator 10. The user holds the encryption device 20 or the decryption device 50 or both. The group administrator holds the re-encryption
15 key generator 30. When, for example, the user administrator or the server Sv generates a re-encryption key, the user administrator or the server Sv holds the re-encryption key generator 30. The server Sv holds the re-encryption device 40. In the file sharing system
20 illustrated in FIG. 8, the server Sv holds the public key storage unit 1 and the user C who performs encryption acquires the public key used for encryption from the public key storage unit 1, but the present embodiment is not
25 limited to such an example and, as will be described later, the user C may acquire the public key from the key generator 10 (that has generated the public key used for

encryption). The server Sv stores the ciphertext $E(pk_{Gr}, m)$ generated by the user C in the ciphertext storage unit 2, but the ciphertext storage unit 2 may be included in the re-encryption device 40 or an external storage device not included in the re-encryption device 40.

Each entity may include a portion or all of the key generator 10, the encryption device 20, the re-encryption key generator 30, the re-encryption device 40, and the decryption device 50. That is, which entity holds which device is not limited to the above example and various variations can be considered. The user administrator and the group administrator may be the same or different. A user may or may not serve also as the user administrator, the group administrator, or both. The user administrator or the group administrator may or may not serve also as a user. Each user device used by each user may be called a first user device, a second user device, ... Similarly, the public key, the private key, and the re-encryption key generation key of each user device may be called the first public key, the first private key, and the first re-encryption key generation key of the first user device, the second public key, the second private key, and the second re-encryption key generation key of the second user device, ...

Also, as shown, for example, in each of FIGS. 9 and 10, each entity may hold each device. In FIG. 9, a server Sv2 holds the key generator 10 to generate a public parameter,

a set of a public key and private key and a re-encryption
key generation key of the user A, and a set of a public key
and private key and a re-encryption key generation key of
the user B. The server Sv2 delivers the public parameter
5 and a set of the public key and private key and the re-
encryption key generation key of the user A to the user A,
the public parameter and a set of the public key and
private key and the re-encryption key generation key of the
user B to the user B, and the public parameter and the
10 public key of the user A to a server Sv1. The user A holds
the encryption device 20, encrypts plaintext data using the
public key of the user A and the public parameter, and
transmits the obtained ciphertext data to the server Sv1.
The server Sv2 holds the re-encryption key generator 30,
15 generates a re-encryption key using the private key of the
user A, the re-encryption key generation key of the user B,
the public parameter, and a random number, and delivers the
obtained re-encryption key to the server Sv1. The server
Sv1 holds the re-encryption device 40, performs
20 verification processing of ciphertext data received from
the user A based on the public key of the user A received
from the server Sv2, the public parameter, and the re-
encryption key, and re-encrypts the ciphertext data
received from the user A without decryption to deliver the
25 obtained re-encrypted data to the user B. The user B holds
the decryption device 50, performs verification processing
of re-encrypted data received from the server Sv1 based on

the private key of the user B and the public parameter, and also performs decryption processing of the re-encrypted data received from the server Sv1 to obtain plaintext data. The exchange of data between each entity is not limited to
5 an electric communication means such as cable communication, wireless communication and the like and, for example, a physical communication means that distributes or delivers by hand a storage medium in which data such as the re-encryption key generation key, re-encryption key,
10 ciphertext data, re-encrypted data and the like is stored may also be used. Alternatively, the electric communication means and the physical communication means may be combined to exchange data between each entity.

Even if only the electric communication means is used,
15 as will be described later, other entities than the relevant entities may be interposed. Consider a case when data (for example, the re-encryption key generation key) is transmitted from some entity (hereinafter, called the first entity) to another entity (hereinafter, called the second
20 entity). The first entity is, for example, the user A. The second entity is, for example, the group administrator. The first entity transmits data to an entity (hereinafter, called the third entity) other than the second entity. The
25 third entity is, for example, the server Sv1. Next, the second entity receives the relevant data from the third entity.

In FIG. 10, the user A holds the key generator 10 to

generate a public parameter, a set of a public key and private key and a re-encryption key generation key of the user A, and a set of a public key and private key and a re-encryption key generation key of the user B. The user A
5 delivers the public parameter and a set of the public key and private key and the re-encryption key generation key of the user B to the user B and the public parameter and the public key of the user A to the server Sv1. The user A holds the encryption device 20, encrypts plaintext data
10 using the public key of the user A and the public parameter, and transmits the obtained ciphertext data to the server Sv1. The user A holds the re-encryption key generator 30, generates a re-encryption key using the private key of the user A, the re-encryption key generation key of the user B,
15 the public parameter, and a random number, and delivers the obtained re-encryption key to the server Sv1. The server Sv1 holds the re-encryption device 40, performs verification processing of ciphertext data received from the user A based on the public key of the user A received
20 from the user A, the public parameter, and the re-encryption key, and re-encrypts the ciphertext data received from the user A without decryption to deliver the obtained re-encrypted data to the user B. The user B holds the decryption device 50, performs verification processing
25 of re-encrypted data received from the server Sv1 based on the private key of the user B and the public parameter, and also performs decryption processing of the re-encrypted

data received from the server Sv1 to obtain plaintext data.

Subsequently, the configuration of each of the devices 10 to 50 will concretely be described.

5 The key generator 10 includes a key generation parameter storage unit 11, a temporary data storage unit 12, a public parameter generation unit 13, a key generation unit 14, a communication unit 15, and a control unit 16.

The key generation parameter storage unit 11 is a storage device that stores a key generation parameter.

10 The temporary data storage unit 12 is a storage device that stores transient data (hereinafter, also called temporary data) such as data being processed by each of the generation units 13, 14 and processing results.

15 The public parameter generation unit 13 generates a public parameter for key generation.

The key generation unit 14 generates a pair of a public key/private key and a re-encryption key generation key.

20 The communication unit 15 is a communication interface to communicate with the other devices 20 to 50 and has, for example, a function to transmit the public parameter and a pair of the public key/private key and the re-encryption key generation key of each of the devices 30, 50 in the temporary data storage unit 12 to each of the relevant
25 devices 30, 50 under the control of the control unit 16, a function to transmit the public key of the re-encryption key generator 30 in the temporary data storage unit 12 to

the encryption device 20 under the control of the control unit 16, and a function to transmit the public parameter and the public key of the re-encryption key generator 30 in the temporary data storage unit 12 to the re-encryption device 40 under the control of the control unit 16. In the description that follows, the description that transmission/reception goes through the communication unit 15 may be omitted, to avoid a redundant description. This also applies to descriptions of communication units of the other devices 20 to 50.

The control unit 16 has a function to control each of the units 11 to 15 such that each unit operates as shown in FIG. 2.

The encryption device 20 includes a temporary data storage unit 21, a communication unit 22, an encrypted data generation unit 23, a ciphertext generation unit 24, and a control unit 25.

The temporary data storage unit 21 is a storage device that stores a public parameter received from the key generator 10 and a public key of the re-encryption key generator 30 and also transient data such as data being processed by each of the generation devices 23, 24 and processing results.

The communication unit 22 is a communication interface to communicate with the other devices 10, 30 to 50 and has, for example, a function to acquire and write the public parameter received from the key generator 10 and the public

key of the re-encryption key generator 30 into the
temporary data storage unit 21 and a function to transmit
ciphertext data in the temporary data storage unit 21 to
the re-encryption device 40 under the control of the
5 control unit 25.

The encryption parameter generation unit 23 has a
function to generate an encryption parameter.

The ciphertext generation unit 24 has a function to
generate ciphertext data by encrypting plaintext data using
10 the public parameter read from the temporary data storage
unit 21 and the public key of the re-encryption key
generator 30 (first public key of the first user device)
and a function to write the obtained ciphertext data into
the temporary data storage unit 21.

15 The control unit 25 has a function to control each of
the units 21 to 24 such that each unit operates as shown in
FIG. 3.

The re-encryption key generator 30 includes a private
key storage unit 31, a temporary data storage unit 32, a
20 communication unit 33, a re-encryption key generation unit
34, a control unit 35, and a random number generation unit
36.

The private key storage unit 31 is a storage device
that stores a private key of the re-encryption key
25 generator 30 received from the key generator 10.

The temporary data storage unit 32 is a storage device
that stores a public parameter received from the key

generator 10 and a re-encryption key generation key of the decryption device 50 and also temporary data such as data being processed by the re-encryption key generation unit 34 and processing results.

5 The communication unit 33 is a communication interface to communicate with the other devices 10, 20, 40, 50 and has, for example, a function to acquire and write the public parameter received from the key generator 10 and the re-encryption key generation key of the decryption device
10 50 into the temporary data storage unit 32 and a function to transmit the re-encryption key in the temporary data storage unit 32 to the re-encryption device 40 under the control of the control unit 35.

 The re-encryption key generation unit 34 has a
15 function to generate a re-encryption key based on the private key of the re-encryption key generator 30 (first private key of the first user device) read from the private key storage unit 31, the public parameter and the re-encryption key generation key of the decryption device 50
20 (second re-encryption key generation key of the second user device) read from the temporary data storage unit 32, and a random number generated by the random number generation unit 36 and a function to write the re-encryption key into the temporary data storage unit 32.

25 The control unit 35 has a function to control each of the units 31 to 34, 36 such that each unit operates as shown in FIG. 4.

The random number generation unit 36 has a function to generate a random number and to send out the random number to the re-encryption key generation unit 34.

5 The re-encryption key generator 30 has a re-encryption key generation key storage unit (not shown) as a storage device that stores a re-encryption key generation key of the local device 30 received from the key generator 10 when the re-encryption key generation key of the local device 30 is received from the key generator 10.

10 The re-encryption device 40 includes a re-encryption key storage unit 41, a temporary data storage unit 42, a communication unit 43, a re-encryption processing unit 44, a re-encryption parameter generation unit 45, and a control unit 46.

15 The re-encryption key storage unit 41 is a storage unit that stores a re-encryption key received from the re-encryption key generator 30.

20 The temporary data storage unit 42 is a storage device that stores a public parameter received from the key generator 10 and a public key of the re-encryption key generator 30 and also temporary data such as data being processed by the re-encryption processing unit 44 and processing results.

25 The communication unit 43 is a communication interface to communicate with the other devices 10 to 30, 50 and has, for example, a function to acquire and write the public parameter received from the key generator 10 and the public

key of the re-encryption key generator 30 into the temporary data storage unit 42, a function to send out ciphertext data received from the encryption device 20 to the re-encryption processing unit 44, and a function to transmit re-encrypted data in the temporary data storage unit 42 to the decryption device 50 under the control of the control unit 46.

The re-encryption processing unit 44 has a function to obtain, when ciphertext data is received from the encryption device 20, re-encrypted data by re-encrypting the ciphertext data without decryption using the re-encryption key read from the re-encryption key storage unit 41 and the public parameter and the public key of the re-encryption key generator 30 read from the temporary data storage unit 42 and a function to write the obtained re-encrypted data into the temporary data storage unit 42.

The control unit 46 has a function to control each of the units 41 to 45 such that each unit operates as shown in FIG. 5.

The decryption device 50 includes a private key storage unit 51, a temporary data storage unit 52, a communication unit 53, a decryption processing unit 54, a re-encryption key generation key storage unit 55, and a control unit 56.

The private key storage unit 51 is a storage device that stores a private key of the local device 50 received from the key generator 10.

The temporary data storage unit 52 is a storage device that stores a public parameter received from the key generator 10 and temporary data such as data being processed by the decryption processing unit 54 and processing results.

The communication unit 53 is a communication interface to communicate with the other devices 10 to 40 and has, for example, a function to acquire and write the private key of the local device 50 received from the key generator 10 into the private key storage unit 51, a function to acquire and write the public parameter received from the key generator 10 into the temporary data storage unit 52, and a function to send out re-encrypted data received from the re-encryption device 40 to the decryption processing unit 54.

The decryption processing unit 54 has a function to obtain plaintext data when re-encrypted data is received from the re-encryption device 40 by performing decryption processing of the re-encrypted data using the private key of the decryption device 50 (second private key of the second user device) read from the private key storage unit 51 and the public parameter read from the temporary data storage unit 52 and a function to write the obtained plaintext data into the temporary data storage unit 52.

The re-encryption key generation key storage unit 55 is a storage device that stores a re-encryption key generation key of the local device 50 received from the key generator 10.

The control unit 56 has a function to control each of the units 51 to 55 such that each unit operates as shown in FIG. 6.

5 In the present embodiment, for example, the public parameter is stored in the temporary data storage unit of each of the devices 10 to 50. However, the present embodiment is not limited to such an example and the public parameter may be stored in a storage unit (separate from the temporary data storage unit) of each of the devices 10
10 to 50. Thus, of data stored in the temporary data storage unit of each of the devices 10 to 50, data that does not have to be handled as transient data may be stored in a storage unit (separately from the temporary data storage unit). When the public parameter includes a plurality of
15 elements, each of the devices 20 to 50 may acquire, of each element of the public parameter, only elements needed for each piece of processing described later. The above also apply to embodiments and modifications described later.

Next, the operation of the re-encryption system
20 configured as described above will be described using the sequence diagrams of FIGS. 2 to 6.

The following operation takes a case in which processing is performed in a procedure of (1) key setup processing, (2) encryption processing, (3) re-encryption
25 key generation processing, (4) re-encryption processing, and (5) decryption processing as an example. However, in the following operation, processing is not necessarily

performed following the above procedure. For example,
processing of re-encryption key generation may be performed
before the encryption processing. Also, ciphertext data
may be decrypted without the re-encryption processing being
5 performed.

(1) The key setup processing is performed by the key
generator 10 as shown in FIG. 2 and each of steps ST1 to
ST12 below.

First, the public parameter generation unit 13 of the
10 key generator 10 generates a public parameter

$$\text{par} = \{\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_2, g_3, g_4, g_5, u, v, \text{Sig}(\hat{G}, \hat{S}, \hat{V})\}$$

based on a security parameter λ stored in the key
generation parameter storage unit 11 in advance or acquires
the public parameter from outside (ST1).

15 More specifically, the public parameter generation
unit 13 selects bilinear map groups $(\mathbb{G}, \mathbb{G}_T)$ satisfying the
order $p > 2\lambda$ based on the security parameter λ stored in the
key generation parameter storage unit 11 in advance. In
this case, p is an integer. p may be a prime number or a
20 composite number. The public parameter generation unit 13
randomly selects elements g, g_1, g_2, u, v of $\mathbb{G}$ from $\mathbb{G}$. The
public parameter generation unit 13 randomly selects π from
 $\mathbb{Z}_p^*$ and calculates $g_3 = g^\pi$, $g_4 = g_1^\pi$, and $g_5 = g_2^\pi$. $\mathbb{Z}_p^*$ is a set
($= (\mathbb{Z}/p\mathbb{Z})^*$) of integers mutually prime to $\mathbb{Z}_p$ and p and may
25 also be called a multiplicative group $\mathbb{Z}_p^*$ for p . $\mathbb{Z}_p$ is a
set ($= (\mathbb{Z}/p\mathbb{Z})$) of integers equal to 0 or greater and less
than p . g_3 is a value based on a random number π and a

system fixed value g . g_4 is a value based on the random number π and a system fixed value g_1 . g_5 is a value based on the random number π and a system fixed value g_2 .

Further, the public parameter generation unit 13 generates a one-time signature algorithm $\text{Sig}(\hat{G}, \hat{S}, \hat{V})$.

$\hat{G}$ in the one-time signature algorithm $\text{Sig}(\hat{G}, \hat{S}, \hat{V})$ (hereinafter, $\text{Sig}(\hat{G}, \hat{S}, \hat{V})$ is also called Sig) means a function to generate a pair (ssk, svk) of a signature key and a verification key (both are one-time keys), $\hat{S}$ means a function to generate a signature σ for a message M , and $\hat{V}$ means a function to validate the signature σ .

For more information about the one-time signature, see A. Menezes, P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography", CRC Press, (1996) pp.462-471, (1996).

The bilinear map (denoted by e) is a map yielding $e: G \times G \rightarrow G_T$ and satisfies the following three properties:

1. $e(g^a, h^b) = e(g, h)^{ab}$ holds for any $(g, h) \in G \times G$ and $a, b \in \mathbb{Z}$, where $\mathbb{Z}$ is a set of integers.

2. $e(g, h)$ is computable for any $(g, h) \in G \times G$.

3. $e(g, h) \neq 1_{G_T}$ always holds if $g, h \neq 1_G$, where 1_G is the unit element of G and 1_{G_T} is the unit element of G_T .

The bilinear map group (denoted by G, G_T) is a group of the order p for which a bilinear map $e: G \times G \rightarrow G_T$ exists. The above definition also means that if $g_1 = g^\alpha$ and $g_2 = g^\beta$ for elements g, g_1, g_2 of G , the following equation holds (α is the exponent relating to elements g, g_1 of G and β is the exponent relating to elements g, g_2 of G). The "exponent

relating to elements g, g_1 of G is an exponent that defines the relationship between elements g, g_1 of G and represents a value α such that the element obtained by the α -th power of g and g_1 are equal in G . Similarly, the
5 "exponent relating to elements g, g_2 of G is an exponent that defines the relationship between elements g, g_2 of G and represents a value β such that the element obtained by the β -th power of g and g_2 are equal in G . As will be described later, G may also be expressed as a
10 multiplicative group and in such a case, if, for example, elements of G are points on an elliptic curve, the exponent may be changed to a scalar, the α -th power to α times, and the β -th power to β times.

$$e(g_1g_2, g) = e(g, g_1g_2) = e(g, g^\alpha \cdot g^\beta) = e(g, g^{\alpha+\beta}) = e(g, g)^{\alpha+\beta} = e(g, g)^\alpha \cdot e(g, g)^\beta$$

15

Incidentally, a plurality of elements g, g_1, g_2 of G is a plurality of preset system fixed values. The plurality of system fixed values is not limited to three elements of the bilinear map group G and a plurality of
20 elements such as two or four or more elements of G can also be used when appropriate. The term "system fixed value" is interchangeable with "fixed value", "element", and "system parameter".

In this specification, notation assuming that both G
25 and G_T are multiplicative groups is adopted, but the embodiments are not limited to such notation and each of G, G_T can be represented by a notation assuming that each is

an additive group. That is, for example, G may be represented as an additive group and G_T as a multiplicative group. For example, elements of G may be points on an elliptic curve and elements of G_T may be elements on a finite field. As the bilinear map, a map that yields $e:G_1 \times G_2 \rightarrow G_T$ for bilinear map groups G_1, G_2, G_T (G_1 and G_2 are different groups) may also be used. In this case, each element of G in each piece of processing described later may be rewritten as element G_1 or element G_2 , where appropriate. The above also apply to embodiments and modifications described later.

Subsequently, the public parameter generation unit 13 writes the generated public parameter par into the temporary data storage unit 12. The key generator 10 publishes the public parameters $par=(\lambda, p, G, G_T, g, g_1, g_2, g_3, g_4, g_5, u, v, Sig)$ in the temporary data storage unit 12 (ST2). If a public parameter is published before step ST1 is executed, step ST1 and step ST2 may be skipped by writing the public parameter into the temporary data storage unit 12.

If the identification information of the re-encryption key generator 30 is i , the key generation unit 14 randomly selects elements x_i, y_i, z_i of Z_p^* from Z_p^* and sets a private key $sk_i=(x_i, y_i, z_i)$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $pk_i=(X_i, Y_{1i}, Y_{2i}, Z_i, Z_{1i})$ of the re-encryption key generator 30 using the private key sk_i and the public

parameter par.

$$X_i = g^{X_i}, \quad Y_{1i} = g_1^{Y_i}, \quad Y_{2i} = g_2^{Y_i}, \\ Z_i = g^{Z_i}, \quad Z_{1i} = g_1^{Z_i}$$

When verification processing using a formula (a2') described later is not performed, data Y_{2i} may be omitted. That is, the public key pk_i only needs to include data X_i , Y_{1i} , Z_i , Z_{1i} .

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i = (X_{3i}, Y_{5i})$ using the private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{X_i}, \quad Y_{5i} = g_5^{Y_i}$$

Subsequently, the key generation unit 14 writes a pair of the generated public key/private key and the re-encryption key generation key into the temporary data storage unit 12. The communication unit 15 transmits the private key sk_i and the re-encryption key generation key rk_i in the temporary data storage unit 12 to the re-encryption key generator 30 under the control of the control unit 16 (ST4). The key generator 10 publishes the public key pk_i of the re-encryption key generator 30 in the temporary data storage unit 12 (ST5).

Similarly, if the identification information of the decryption device 50 is j , the key generation unit 14 randomly selects elements x_j , y_j , z_j of Z_p^* from Z_p^* and sets a private key $sk_j = (x_j, y_j, z_j)$ of the decryption device 50.

Then, the key generation unit 14 generates a public key $pk_j = (X_j, Y_{1j}, Y_{2j}, Z_j, Z_{1j})$ of the decryption device 50 using the private key sk_j and the public parameter par .

$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Y_{2j} = g_2^{y_j}, \\ Z_j = g^{z_j}, \quad Z_{1j} = g_1^{z_j}$$

5 If, similarly as described above, the verification processing using a formula (a2') described later is not performed, data Y_{2j} may be omitted. That is, the public key pk_j only needs to include data X_j, Y_{1j}, Z_j, Z_{1j} .

10 Further, the key generation unit 14 generates a re-encryption key generation key $rk_j = (X_{3j}, Y_{5j})$ using the private key sk_j and the public parameter par (ST6).

$$X_{3j} = g_3^{x_j}, \quad Y_{5j} = g_5^{y_j}$$

15 Subsequently, the key generation unit 14 writes a pair of the generated public key/private key and the re-encryption key generation key into the temporary data storage unit 12. The communication unit 15 transmits the private key sk_j and the re-encryption key generation key rk_j in the temporary data storage unit 12 to the decryption device 50 under the control of the control unit 16 (ST7).

20 The key generator 10 publishes the public key pk_j of the decryption device 50 in the temporary data storage unit 12 (ST8). If desired, processing similar to steps ST6 to ST8 may be performed on a private key sk_h , a public key pk_h , and a re-encryption key generation key rk_h of the

encryption device 20 to transmit the private key sk_h and the re-encryption key generation key rk_h to the encryption device 20 and to publish the public key pk_h . The key generator 10 may transmit only the private key (without transmitting the re-encryption key generation key) to the encryption device 20, the re-encryption key generator 30, or the decryption device 50. The key generator 10 may also transmit only a portion or all of public keys generated in key setup processing to the encryption device 20, the re-encryption key generator 30, or the decryption device 50. The above also apply to embodiments and modifications described later.

With the above processing, the key setup processing is completed. Hereinafter, each of the devices 20, 30, 40, 50 can acquire and use the public parameter or public key published in steps ST2, ST5, ST8 when appropriate.

(2) The encryption processing is performed by the encryption device 20 as shown in FIG. 3 and each of steps ST21 to ST24 below. The encryption device 20 is assumed to have a public key pk_i of the re-encryption key generator 30 stored in the temporary data storage unit 21, but if the encryption device 20 has not yet acquired the public key pk_i of the re-encryption key generator 30, the encryption device 20 may acquire the public key pk_i of the re-encryption key generator 30 from, for example, the key generator 10 before step ST 22 described below. Similarly, the encryption device 20 is assumed to have the public

parameter par stored in the temporary data storage unit 21,
but if the encryption device 20 has not yet acquired the
public parameter par, the encryption device 20 may acquire
the public parameter par from, for example, the key
5 generator 10 before step ST 21 described below.

The encryption parameter generation unit 23 of the
encryption device 20 generates a key pair $(ssk,svk)=\hat{G}(\lambda)$
of a signature key ssk and a verification key svk in a one-
time signature using the security parameter λ and a key
10 pair generation function $\hat{G}$ in the public parameter par
(ST21) and sets the verification key svk to encrypted data
 C_1 ($C_1=svk$).

The encryption parameter generation unit 23 also
generates and sends out a random number $r \in \mathbb{Z}_p^*$ to the
15 ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted
data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4$ below from a message $m \in G_T$ as
plaintext data using the random number r , the public key
 pk_i of the re-encryption key generator 30, and the public
20 parameter par (ST22).

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r,$$
$$C_3 = e(g_4 g_5, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

After step ST22 is completed, the ciphertext
generation unit 24 generates a one-time signature σ as
shown in the following formula for the encrypted data C_3 ,
25 C_4 based on a signature generation function $\hat{S}$ of the

public parameter par and the signature key ssk generated in step ST21 (ST23).

$$\sigma = \hat{S}(ssk, (C_3, C_4))$$

The first argument of the function $\hat{S}$ is the
5 signature key and the second argument is data to be signed.

The function $\hat{S}$ accepts the signature key and data to be signed as input and outputs a signature.

Thereafter, the ciphertext generation unit 24
generates ciphertext data $C_i = (C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, \sigma)$
10 including all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext data into the temporary data storage unit 21.

If the verification processing in step ST41 described later is omitted, the ciphertext data
15 $C_i = (C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, \sigma)$ may be set as $C_i = (C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)$ by omitting verification data (C_1, C_4, σ) that is not used for decryption. In this case, processing to generate the verification data (C_1, C_4, σ) is also omitted. In the present embodiment, as described
20 above (or will be described below), verification may be omitted or generation of verification data may be omitted and this also applies to embodiments and modifications described below.

In any case, the communication unit 22 transmits the
25 ciphertext data C_i in the temporary data storage unit 21 to the re-encryption device 40 under the control of the control unit 25 (ST24).

With the above processing, the encryption processing is completed.

(3) The re-encryption key generation processing is performed by the re-encryption key generator 30 as shown in FIG. 4 and each of steps ST31 to ST33 below. The re-encryption key generator 30 is assumed to have the public parameter par stored in the temporary data storage unit 32, but if the re-encryption key generator 30 has not yet acquired the public parameter par , the re-encryption key generator 30 may acquire the public parameter par from, for example, the key generator 10 before step ST 32 described below.

Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. The communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into the temporary data storage unit 32 under the control of the control unit 35 (ST31). In step ST31, the key generator 10

(instead of the decryption device 50) may transmit the re-encryption key generation key rk_j to the re-encryption key generator 30. This also applies to embodiments and modifications described later.

5 Regarding the temporary data storage unit 32, as shown in FIG. 11, the re-encryption key generator 30 may hold a public parameter storage unit 321 and a re-encryption key generation key storage unit 322 in the temporary data storage unit 32 to store the public parameter par in the
10 public parameter storage unit 321 and the re-encryption key generation key rk_j of the decryption device 50 in the re-encryption key generation key storage unit 322. The public parameter storage unit 321 that stores the public parameter par may be outside the temporary data storage unit 32.
15 Also, the re-encryption key generation key storage unit 322 that stores the re-encryption key generation key rk_j may be outside the temporary data storage unit 32. The above also apply to embodiments and modifications described later. In step ST4 described above, the communication unit 33
20 receives the private key sk_i of the re-encryption key generator 30 from the key generator 10 and writes the private key sk_i into the private key storage unit 31.

 The random number generation unit 36 generates and sends out a random number $\theta \in \mathbb{Z}_p^*$ to the re-encryption key
25 generation unit 34.

 The re-encryption key generation unit 34 generates a re-encryption key R_{ij} as shown in the following formulas

using the random number θ , the private key sk_i of the re-encryption key generator 30 in the private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, and the public parameter par (ST32).

$$R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$$

$$R_{ij1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} = g^{\frac{\pi(x_j + \theta)}{x_i}},$$

$$R_{ij2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} = g^{\frac{\pi(\beta y_j + \theta)}{y_i}},$$

$$R_{ij3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j + \beta y_j + \theta)}{z_i}}.$$

In the above formulas, g_2 is rewritten and denoted by g^β .

Thereafter, the re-encryption key generation unit 34 writes the generated re-encryption key R_{ij} into the temporary data storage unit 32. The communication unit 33 transmits the re-encryption key R_{ij} in the temporary data storage unit 32 to the re-encryption device 40 under the control of the control unit 35 (ST33).

With the above processing, the re-encryption key generation processing is completed.

(4) The re-encryption processing is performed by the re-encryption device 40 as shown in FIG. 5 and each of steps ST41 to ST43 below. The re-encryption device 40 is

assumed to have a public key pk_i of the re-encryption key generator 30 stored in the temporary data storage unit 42, but if the re-encryption device 40 has not yet acquired the public key pk_i of the re-encryption key generator 30, the re-encryption device 40 may acquire the public key pk_i of the re-encryption key generator 30 from, for example, the re-encryption key generator 30 before step ST 41 described below. Similarly, the re-encryption device 40 is assumed to have the public parameter par stored in the temporary data storage unit 42, but if the re-encryption device 40 has not yet acquired the public parameter par , the re-encryption device 40 may acquire the public parameter par from, for example, the key generator 10 before step ST 41 described below.

The communication unit 43 of the re-encryption device 40 writes the ciphertext data C_i transmitted in step ST24 and the re-encryption key R_{ij} transmitted in step ST33 into the temporary data storage unit 42.

The re-encryption processing unit 44 verifies the ciphertext data C_i in the temporary data storage unit 42 using the public parameter par and the public key pk_i of the re-encryption key generator 30 based on the following verification formula (ST41).

$$\begin{aligned}e(C_{2X}, u^{C_1} \cdot v) &= e(X_i, C_4), \\e(C_{2Y}, u^{C_1} \cdot v) &= e(Y_{1i}, C_4), \\e(C_{2Z}, u^{C_1} \cdot v) &= e(Z_i, C_4), \\e(C_{2Z1}, u^{C_1} \cdot v) &= e(Z_{1i}, C_4), \\ \hat{V}(C_1, \sigma, (C_3, C_4)) &= 1.\end{aligned}$$

... (a1)

The first argument of the function $\hat{V}$ is a verification key, the second argument is a signature, and the third argument is data to be signed.

5 The function $\hat{V}$ verifies the signature for the data to be verified using the verification key and outputs "1" when the signature is determined to be valid and any value other than "1" (for example, "0") when the signature is determined not to be valid.

10 Therefore, the fifth verification formula in the formula (a1) shows that the one-time signature σ is valid by verifying the signature σ for the encrypted data C_3, C_4 as data to be signed by using the encrypted data C_1 as the verification key.

15 The verification is successful if the five verification formulas in the formula (a1) all hold and fails if any one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST41 is skipped), the re-encryption
20 parameter generation unit 45 generates and sends out three random numbers $s, t, k \in \mathbb{Z}_p^*$ to the re-encryption processing

unit 44.

The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' , C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{5X} , C_{5Y} , C_{5Z} using these random numbers s , t , k , the ciphertext data C_i stored in the temporary data storage unit 42, the re-encryption key R_{ij} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$\begin{aligned} C_{2X}' &= X_i^s, C_{2X}'' = C_{2X}'^s = X_i^{rs}, C_{2Y}' = Y_{1i}^t, C_{2Y}'' = C_{2Y}'^t = Y_{1i}^{rt}, \\ C_{2Z}' &= Z_i^k, C_{2Z}'' = C_{2Z}'^k = Z_i^{rk}, C_{2Z1}' = Z_{1i}^k, C_{2Z1}'' = C_{2Z1}'^k = Z_{1i}^{rk}, \\ C_{5X} &= R_{ij1}^{\frac{1}{s}}, C_{5Y} = R_{ij2}^{\frac{1}{t}}, C_{5Z} = R_{ij3}^{\frac{1}{k}} \end{aligned}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data $C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4, \sigma)$ by replacing the encrypted data C_{2X} , C_{2Y} , C_{2Z} , C_{2Z1} in the ciphertext data C_i with all the above re-encrypted data C_{2X}' to C_{5Z} and writes the obtained re-encrypted data C_j into the temporary data storage unit 42. If verification processing in step ST51 in decryption processing described later is omitted, the re-encrypted data $C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4, \sigma)$ may be set as $C_j = (C_{2X}'', C_{2Y}'', C_{2Z}'', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3)$ by omitting verification data $(C_1, C_{2X}', C_{2Y}', C_{2Z}', C_{2Z1}', C_4, \sigma)$ not used for decryption. In this case, processing to generate the verification data $(C_{2X}', C_{2Y}', C_{2Z}', C_{2Z1}')$ is also omitted.

In any case, the communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of the control unit 46 (ST43).

5 With the above processing, the re-encryption processing is completed.

 Incidentally, the re-encryption processing unit 44 may perform the re-encrypted data generation processing in step ST42 by omitting the verification processing in step ST41.
10 When verification data (C_1, C_4, σ) is not contained in the ciphertext data C_i , the re-encryption processing unit 44 performs the re-encrypted data generation processing in step ST42 by omitting the verification processing in step ST41.

15 (5) The decryption processing is performed by the decryption device 50 as shown in FIG. 6 and each of steps ST51 and ST52 below. The decryption device 50 is assumed to have the public parameter par stored in the temporary data storage unit 52, but if the decryption device 50 has
20 not yet acquired the public parameter par , the decryption device 50 may acquire the public parameter par from, for example, the key generator 10 before step ST 51 described below.

 The communication unit 53 of the decryption device 50
25 receives the re-encrypted data C_j transmitted in step ST43 and writes the re-encrypted data C_j into the temporary data storage unit 52.

The decryption processing unit 54 verifies the re-encrypted data C_j in the temporary data storage unit 52 using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 based on the following verification formulas (ST51).

$$\begin{aligned}
 e(C''_{2X}, u^{C_1} \cdot v) &= e(C'_{2X}, C_4), \\
 e(C''_{2Y}, u^{C_1} \cdot v) &= e(C'_{2Y}, C_4), \\
 e(C''_{2Z}, u^{C_1} \cdot v) &= e(C'_{2Z}, C_4), \\
 e(C''_{2Z1}, u^{C_1} \cdot v) &= e(C'_{2Z1}, C_4), \\
 e(C_{5Z}, C'_{2Z}) &= e(C_{5X}, C'_{2X}) \cdot e(g_2, g_3)^{y_j}, \\
 e(C_{5Z}, C'_{2Z1}) &= e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j}, \\
 \hat{V}(C_1, \sigma, (C_3, C_4)) &= 1.
 \end{aligned}
 \tag{a2}$$

Instead of the fifth verification formula in the formula (a2), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) = e(C_{5X}, C'_{2X}) \cdot e(g_5, g)^{y_j}.$$

Also, instead of the fifth verification formula in the formula (a2), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) = e(C_{5X}, C'_{2X}) \cdot e(Y_{2j}, g_3). \tag{a2'}$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

Also, instead of the sixth verification formula in the

formula (a2), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(g_3, g_1)^{x_j}.$$

Also, instead of the sixth verification formula in the formula (a2), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4).$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

The verification is successful if the seven verification formulas in the formula (a2) all hold and fails if any one of the seven formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned}
& C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\
&= C_3 / \left\{ \left(\frac{e(g^{\pi(x_j + \beta y_j + \theta)/z_i k}, g^{z_i r k})}{e(g^{\pi(x_j + \theta)/x_i s}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi(x_j + \beta y_j + \theta)/z_i k}, g^{\alpha z_i r k})}{e(g^{\pi(\beta y_j + \theta)/y_i t}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \\
&= C_3 / \left\{ \left(\frac{e(g^{\pi(x_j + \beta y_j + \theta)}, g^r)}{e(g^{\pi(x_j + \theta)}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi(x_j + \beta y_j + \theta)}, g^{\alpha r})}{e(g^{\pi(\beta y_j + \theta)}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\
&= m \cdot e(g_4 g_5, g)^r / (e(g, g)^{\pi \beta r} \cdot e(g, g)^{\pi \alpha r}) = m.
\end{aligned}$$

5 With the above processing, the decryption processing is completed.

Incidentally, the decryption processing unit 54 may perform the decryption processing in step ST52 by omitting the verification processing in step ST51. Also if the re-encrypted data C_j does not contain the verification data $(C_1, C_{2X}', C_{2Y}', C_{2Z}', C_{2Z1}', C_4, \sigma)$, the decryption processing unit 54 performs the decryption processing in step ST52 by omitting the verification processing in step ST51. The above m may be a key, instead of a message itself. For example, a shared key key in shared key cryptography may be set as $m = \text{key}$ to replace m with key. In this case, ciphertext obtained by encrypting a message using the shared key may be added to the ciphertext data or re-encrypted data. These also apply to embodiments and modifications described later.

10

15

20

In the present embodiment, the order of processing may be changed when appropriate. For example, the order of the decryption processing and the ciphertext verification processing may be changed. Similarly, the order of the re-
5 encryption key generation processing may be changed to before the encryption processing.

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ij} based on the re-encryption key
10 generation key rk_j , the re-encryption key will not be generated without permission of the user of the decryption device 50 (second user device). In other words, a re-encryption key cannot be generated if the transfer destination (in the example of FIG. 8, for example, the
15 user A) does not notify the transfer source (in the example of FIG. 8, the group administrator) of a re-encryption key generation key. Thus, re-encryption (that is, the transfer of decryption right of ciphertext to the relevant transfer destination) can be prevented when there is no approval of
20 the transfer destination and therefore, safe file sharing can be realized.

Also according to the present embodiment, due to the configuration of generating the re-encryption key R_{ij} based on a random number θ , even if a server and a user collude
25 with each other, the decryption right can be prevented from being re-transferred without permission of the transfer source. Accordingly, there is no need to demand extreme

reliability from the server and a file sharing system the user can safely use can be provided.

To supplement the above description, there has been a problem that, if the server operator, the user B, and the user E collude, a third party can calculate a normal re-encryption key (R_{AE}) to re-encrypt ciphertext for the user A into ciphertext for the user E by using a re-encryption key (R_{AB}) to re-encrypt the ciphertext for the user A into ciphertext for the user B, a private key (sk_B) of the user B, and a private key (sk_E) of the user E without permission of the user A as the transfer source (that is, the decryption right is re-transferred without permission of the transfer source).

According to the present embodiment, however, due to the configuration of generating the re-encryption key R_{ij} based on the random number θ , even if the server operator, the user B, and the user E collude, the decryption right can be prevented from being re-transferred without permission of the transfer source.

[First modification]

In the first embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is cited, but the first embodiment may be changed to a form in which the ciphertext data is decrypted without being re-

encrypted. In this case, only the key setup processing,
encryption processing, and decryption processing need to be
performed. The key setup processing in the present
modification is the same as the key setup processing in the
5 first embodiment. Hereinafter, the encryption processing
and the decryption processing in the present modification
will be described.

The difference between the encryption processing in
the present modification and the encryption processing in
10 the first embodiment is only in the last step. To provide
a description by making use of the above symbols, the
identification information of the decryption device 50 is
assumed to be i for the sake of convenience. In this case,
the communication unit 22 of the encryption device 20
15 transmits the ciphertext data C_i in the temporary data
storage unit 21 to the decryption device 50 under the
control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data
 C_i generated by the encryption device 20 using the public
20 parameter par and the private key sk_i of the local device
50 based on the following verification formulas (ST51').

$$\begin{aligned} e(C_{2X}, u^{C_1} \cdot v) &= e(g, C_4)^{x_i}, \\ e(C_{2Y}, u^{C_1} \cdot v) &= e(g_1, C_4)^{y_i}, \\ e(C_{2Z}, u^{C_1} \cdot v) &= e(g, C_4)^{z_i}, \\ e(C_{2Z1}, u^{C_1} \cdot v) &= e(g_1, C_4)^{z_i}, \\ \hat{V}(C_1, \sigma, (C_3, C_4)) &= 1. \end{aligned} \quad \dots (a3)$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

The verification is successful if the five verification formulas in the formula (a3) all hold and fails if any one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_4 g_5, C_{2X})^{\frac{1}{x_i}}$$

The above formula may calculate with

$$m = \frac{C_3}{e\left((g_4g_5)^{\frac{1}{x_i}}, C_{2X}\right)_{\text{using}} (g_4g_5)^{\frac{1}{x_i}}}$$

calculated in advance.

Accordingly, calculation processing loads for decryption can be reduced. This also applies to
5 embodiments and modifications described later.

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} m \cdot e(g_4g_5, g)^r / e(g_4g_5, C_{2X})^{\frac{1}{x_i}} \\ = m \cdot e(g_4g_5, g)^r / e(g_4g_5, g^{x_i r})^{\frac{1}{x_i}} \\ = m \cdot e(g_4g_5, g)^r / e(g_4g_5, g)^r = m. \end{aligned}$$

10 [Second modification]

In addition to the first modification, as will be described below, the first embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup
15 processing, encryption processing, and decryption processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the first embodiment respectively.

20 Hereinafter, the encryption processing and the decryption

processing in the present modification will be described.
In the present modification, the identification information
of the decryption device 50 is assumed to be j .

Like in step ST21, the encryption parameter generation
5 unit 23 of the encryption device 20 generates (ssk, svk)
(ST21') and sets the verification key svk to the encrypted
data C_1 ($C_1=svk$).

The encryption parameter generation unit 23 also
generates and sends out five random numbers $r, s, t, k, \theta \in \mathbb{Z}_p^*$
10 to the ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted
data $C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X},$
 C_{5Y}, C_{5Z}, C_3, C_4 below from a message $m \in G_T$ as plaintext data
using these random numbers r, s, t, k, θ , the public key pk_j of
15 the decryption device 50, and the public parameter (ST22').

$$\begin{aligned} C_{2X}' &= Y_{2j}^s, C_{2X}'' = Y_{2j}^{rs}, C_{2Y}' = X_j^t, C_{2Y}'' = X_j^{rt}, \\ C_{2Z}' &= Y_{2j}^k, C_{2Z}'' = Y_{2j}^{rk}, C_{2Z1}' = X_j^k, C_{2Z1}'' = X_j^{rk}, \\ C_3 &= e(g_4 g_5, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r, \\ C_{5X} &= (g_4 \cdot g_3^\theta)^{\frac{1}{s}}, C_{5Y} = g_3^{\frac{\theta+1}{t}}, C_{5Z} = (g_4 \cdot g_3^{\theta+1})^{\frac{1}{k}} \end{aligned}$$

After step ST22' is completed, like in step ST23, the
ciphertext generation unit 24 generates a one-time
signature σ (ST23').

20 Thereafter, the ciphertext generation unit 24
generates ciphertext data $C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'',$
 $C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2Y}''', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4, \sigma)$ including

all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext data into the temporary data storage unit 21.

5 The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

10 The decryption device 50 verifies the ciphertext data C_j generated by the encryption device 20 in the same manner as in step ST51 (ST51'). If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j of the local device 50 and the public parameter par to obtain
15 the message m (ST52').

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

20 In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

$$\begin{aligned}
 &= C_3 / \left\{ \left(\frac{e(g^{\pi(1+\alpha+\theta)/k}, g^{\beta y_j r k})}{e(g^{\pi(\alpha+\theta)/s}, g^{\beta y_j r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi(1+\alpha+\theta)/k}, g^{x_j r k})}{e(g^{\pi(\theta+1)/t}, g^{x_j r t})} \right)^{\frac{1}{x_j}} \right\} \\
 &= C_3 / \left\{ \left(\frac{e(g^{\pi(1+\alpha+\theta)}, g^{\beta r})}{e(g^{\pi(\alpha+\theta)}, g^{\beta r})} \right) \cdot \left(\frac{e(g^{\pi(1+\alpha+\theta)}, g^r)}{e(g^{\pi(\theta+1)}, g^r)} \right) \right\} \\
 &= m \cdot e(g_4 g_5, g)^r / (e(g, g)^{\pi \beta r} \cdot e(g, g)^{\pi \alpha r}) = m.
 \end{aligned}$$

[Third modification]

5 In the first embodiment, an example in which the public parameter includes three elements g, g_1, g_2 of system fixed values is described, but the first embodiment is not limited to such an example and the element g_2 may not be generated and the element g_2 may not be included in the public parameter. In this case, $g_2=g$ may be set to replace g_2 with g in the first embodiment. This also applies to embodiments and modifications described later.

[Fourth modification]

15 In the first embodiment, an example in which the public parameter includes three elements g, g_1, g_2 of system fixed values is described, but the first embodiment is not limited to such an example and four or more elements of system fixed values may be included. If, for example, four elements g, g_1, g_2, g_6 are included in the public parameter, $g_2=g_2 g_6$ may be set to replace g_2 with $g_2 g_6$ in the first embodiment. If, for example, five elements $g, g_1, g_2,$

g_6, g_7 are included in the public parameter, $g_1=g_1g_6$ and $g_2=g_2g_7$ may be set to replace g_1 with g_1g_6 and g_2 with g_2g_7 in the first embodiment. This also applies to embodiments and modifications described later.

5 [Fifth modification]

In the first embodiment, an example in which one set of (g_3, g_4, g_5) is included in the public parameter is described, but the first embodiment is not limited to such an example and a plurality of sets of (g_3, g_4, g_5) may be included in the public parameter. This also applies to
10 embodiments and modifications described later.

Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present modification will be described and portions common
15 to the first embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

The public parameter generation unit 13 of the key generator 10 randomly selects $\pi_1, \pi_2, \dots, \pi_N$ from Z_p^* and
20 calculates $A_1, A_2, \dots, A_N$ shown in the following formula.

$$A_n = (\text{tag}_n, g_{3,n}, g_{4,n}, g_{5,n}) \quad (n = 1, 2, \dots, N)$$

$$g_{3,n} = g^{\pi_n}, g_{4,n} = g_1^{\pi_n}, g_{5,n} = g_2^{\pi_n}$$

N is a natural number. tag_n ($n=1, 2, \dots, N$) is the identification information of A_n . tag_n may be a numeric value or a character string. Alternatively, an element
25

itself (other than tag_n) may be used as tag_n . In this case, there is no need to add tag_n as an element of A_n and processing to generate tag_n does not need to be performed.

5 The public parameter generation unit 13 generates a public parameter

$$\text{par} = \{\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_2, \{A_1, A_2, \dots, A_N\}, u, v, \text{Sig}(\hat{G}, \hat{S}, \hat{V})\}$$

based on a security parameter λ stored in the key parameter storage unit 11 in advance or acquires the public parameter from outside (ST1').

10 The key generator 10 publishes the public parameter $\text{par}=(\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_2, A_1, A_2, \dots, A_N, u, v, \text{Sig})$ in the temporary data storage unit 12 (ST2').

If the identification information of the re-encryption key generator 30 is assumed to be i , the key generation unit 14 generates a private key $x_i, y_i, z_i \in \mathbb{Z}_p^*$ of the re-encryption key generator 30 and generates a public key $\text{pk}_i=(X_i, Y_{1i}, Y_{2i}, Z_i, Z_{1i})$ of the re-encryption key generator 30 using the private key $\text{sk}_i=(x_i, y_i, z_i)$ and the public parameter par .

$$\begin{aligned} X_i &= g^{x_i}, & Y_{1i} &= g_1^{y_i}, & Y_{2i} &= g_2^{y_i}, \\ Z_i &= g^{z_i}, & Z_{1i} &= g_1^{z_i} \end{aligned}$$

20 When verification processing using a formula (a7') described later is not performed, data Y_{2i} may be omitted. That is, the public key pk_i only needs to include data X_i, Y_{1i}, Z_i, Z_{1i} .

25 Further, the key generation unit 14 generates re-

encryption key generation keys $rk_{i,1}, rk_{i,2}, \dots, rk_{i,N}$ shown by the following formula using the private key sk_i and the public parameter par (ST3').

$$rk_{i,n} = (X_{3i,n}, Y_{5i,n}) \quad (n = 1, 2, \dots, N)$$

$$X_{3i,n} = g_{3,n}^{x_i}, \quad Y_{5i,n} = g_{5,n}^{y_i}$$

5

The communication unit 15 transmits the private key sk_i and the re-encryption key generation keys $rk_{i,1}, rk_{i,2}, \dots, rk_{i,N}$ in the temporary data storage unit 12 to the re-encryption key generator 30 under the control of the control unit 16 (ST4'). Incidentally, the communication unit 15 may transmit, instead of $rk_{i,1}, rk_{i,2}, \dots, rk_{i,N}$, $(tag_1, rk_{i,1}), (tag_2, rk_{i,2}), \dots, (tag_N, rk_{i,N})$ to the re-encryption key generator 30.

Similarly, if the identification information of the decryption device 50 is assumed to be j , the key generation unit 14 generates a private key $sk_j = (x_j, y_j, z_j)$ of the decryption device 50 and generates a public key $pk_j = (X_j, Y_{1j}, Y_{2j}, Z_j, Z_{1j})$ of the decryption device 50 using the private key sk_j .

$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Y_{2j} = g_2^{y_j},$$
$$Z_j = g^{z_j}, \quad Z_{1j} = g_1^{z_j}$$

20

If, similarly as described above, the verification processing using a formula (a7') described later is not performed, data Y_{2j} may be omitted. That is, the public key pk_j only needs to include data X_j, Y_{1j}, Z_j, Z_{1j} .

Further, the key generation unit 14 generates re-encryption key generation keys $rk_{j,1}, rk_{j,2}, \dots, rk_{j,N}$ shown by the following formula using the private key sk_j and the public parameter par (ST6').

$$rk_{j,n} = (X_{3j,n}, Y_{5j,n}) \quad (n = 1, 2, \dots, N)$$
$$X_{3j,n} = g_{3,n}^{x_j}, \quad Y_{5j,n} = g_{5,n}^{y_j}$$

The communication unit 15 transmits the private key sk_j and the re-encryption key generation keys $rk_{j,1}, rk_{j,2}, \dots, rk_{j,N}$ in the temporary data storage unit 12 to the decryption device 50 under the control of the control unit 16 (ST7'). Instead of $rk_{j,1}, rk_{j,2}, \dots, rk_{j,N}$, the communication unit 15 may transmit, $(tag_1, rk_{j,1}), (tag_2, rk_{j,2}), \dots, (tag_N, rk_{j,N})$ to the decryption device 50.

(2) Encryption processing

Like in step ST21 in the first embodiment, the encryption parameter generation unit 23 of the encryption device 20 generates a key pair (ssk, svk) of the signature key ssk and the verification key svk in the one-time signature (ST21') and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

The ciphertext generation unit 24 generates one element from $A_1, A_2, \dots, A_N$ as elements of the public parameter par . It is assumed below that A_n is selected.

The ciphertext generation unit 24 generates encrypted data $C_{2x}, C_{2y}, C_{2z}, C_{2z1}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random number r , the public key

pk_i of the re-encryption key generator 30, and the public parameter par (ST22').

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_3 = e(g_{4,n}g_{5,n}, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

After step ST22 is completed, the ciphertext generation unit 24 generates a one-time signature σ as shown in the following formula for the identification information tag_n of A_n and the encrypted data C₃, C₄ based on a signature generation function $\hat{S}$ of the public parameter par and the signature key ssk generated in step ST21' (ST23').

$$\sigma = \hat{S}(ssk, (tag_n, C_3, C_4))$$

Incidentally, the ciphertext generation unit 24 may generate the one-time signature σ for the encrypted data C₃, C₄ as data to be signed without including the identification information tag_n of A_n. This also applies to embodiments and modifications described later.

Thereafter, the ciphertext generation unit 24 generates ciphertext data C_i=(C₁, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C₃, C₄, σ , tag_n) including all the above encrypted data C₁ to C₄, the one-time signature σ , and the identification information tag_n of A_n.

If the verification processing in step ST41' and step ST51' described later is omitted, the ciphertext data C_i=(C₁, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C₃, C₄, σ) may be set as C_i=(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C₃) by omitting verification data

($C_1, C_4, \sigma, \text{tag}_n$) that is not used for decryption. In this case, processing to generate the verification data (C_1, C_4, σ) is also omitted. However, if decryption processing in a sixth modification described later is performed, the ciphertext generation unit 24 adds the identification information tag_n of A_n to the ciphertext data C_i .

In any case, the communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to the re-encryption device 40 under the control of the control unit 25 (ST24').

(3) Re-encryption key generation processing

Assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) and encrypted by using A_n as an element of the public parameter par is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext encrypted by using A_n to the user A). The decryption device 50 transmits the identification information tag_n of A_n and a re-encryption key generation key $\text{rk}_{j,n}$ of the decryption device 50 to the re-encryption key generator 30. The communication unit 33 of the re-encryption key generator 30 acquires and writes the identification information tag_n of A_n and the re-encryption

key generation key $rk_{j,n}$ of the decryption device 50 into the temporary data storage unit 32 under the control of the control unit 35 (ST31').

5 The re-encryption key generation unit 34 generates a re-encryption key $R_{ij,n}$ as shown in the following formulas using the random number θ , the private key sk_i of the re-encryption key generator 30 in the private key storage unit 31, the identification information tag_n of A_n and the re-encryption key generation key $rk_{j,n}$ of the decryption
10 device 50 in the temporary data storage unit 32, and the public parameter par (ST32').

$$R_{ij,n} = (R_{ij1,n}, R_{ij2,n}, R_{ij3,n})$$

$$R_{ij1,n} = (X_{3j,n} \cdot g_{3,n}^{\theta})^{1/x_i} = g^{\frac{\pi_n(x_j + \theta)}{x_i}},$$

$$R_{ij2,n} = (Y_{5j,n} \cdot g_{3,n}^{\theta})^{1/y_i} = g^{\frac{\pi_n(\beta y_j + \theta)}{y_i}},$$

$$15 \quad R_{ij3,n} = (X_{3j,n} \cdot Y_{5j,n} \cdot g_{3,n}^{\theta})^{1/z_i} = g^{\frac{\pi_n(x_j + \beta y_j + \theta)}{z_i}}.$$

In the above formulas, g_2 is rewritten and denoted by g^{β} .

(4) Re-encryption processing

The re-encryption processing unit 44 verifies the
20 ciphertext data C_i in the temporary data storage unit 42 using the public parameter par and the public key pk_i of

the re-encryption key generator 30 based on the following verification formula (ST41').

$$\begin{aligned} e(C_{2X}, u^{C_1} \cdot v) &= e(X_i, C_4), \\ e(C_{2Y}, u^{C_1} \cdot v) &= e(Y_{1i}, C_4), \\ e(C_{2Z}, u^{C_1} \cdot v) &= e(Z_i, C_4), \\ e(C_{2Z1}, u^{C_1} \cdot v) &= e(Z_{1i}, C_4), \\ \hat{V}(C_1, \sigma, (\text{tag}_n, C_3, C_4)) &= 1. \end{aligned} \quad \dots (a4)$$

5 If the identification information tag_n of A_n is not included as data to be signed by the one-time signature σ , the fifth verification formula in the formula (a4) conducts verification of the one-time signature σ by restricting data to be signed to the encrypted data C_3, C_4 . This also applies to verification processing of the one-time
10 signature σ described later.

The verification is successful if the five verification formulas in the formula (a4) all hold and fails if any one of the five formulas does not hold.

15 If the verification is successful (or the verification processing in step ST41' is skipped), the re-encryption parameter generation unit 45 generates and sends out three random numbers $s, t, k \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

20 The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' ,

$C_{2X}''', C_{2Y}', C_{2Y}''', C_{2Z}', C_{2Z}''', C_{2Z1}', C_{2Z1}''', C_{5X}, C_{5Y}, C_{5Z}$
using these random numbers s, t, k , the ciphertext data C_i
stored in the temporary data storage unit 42, the re-
encryption key $R_{ij,n}$, the public key pk_i of the re-
5 encryption key generator 30, and the public parameter par
(ST42').

$$C'_{2X} = X_i^s, C''_{2X} = C_{2X}^s = X_i^{rs}, C'_{2Y} = Y_{1i}^t, C''_{2Y} = C_{2Y}^t = Y_{1i}^{rt}, \\ C'_{2Z} = Z_i^k, C''_{2Z} = C_{2Z}^k = Z_i^{rk}, C'_{2Z1} = Z_{1i}^k, C''_{2Z1} = C_{2Z1}^k = Z_{1i}^{rk}, \\ C_{5X} = R_{ij1,n}^{\frac{1}{s}}, C_{5Y} = R_{ij2,n}^{\frac{1}{t}}, C_{5Z} = R_{ij3,n}^{\frac{1}{k}}$$

Incidentally, the re-encryption processing unit 44 may
10 select the re-encryption key $R_{ij,n}$ to be used in step ST42'
based on the identification information tag_n of A_n
contained in the ciphertext data C_i transmitted in step
ST24'.

After step ST42' is completed, the re-encryption
15 processing unit 44 replaces encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}$
in the ciphertext data C_i with all the above re-encrypted
data C_{2X}' to C_{5Z} and further adds the identification
information tag_n of A_n contained in the ciphertext data C_i
transmitted in step ST24' to the re-encrypted data C_j .

20 That is,

$C_j = (C_1, C_{2X}', C_{2X}''', C_{2Y}', C_{2Y}''', C_{2Z}', C_{2Z}''', C_{2Z1}', C_{2Z1}''', C_{5X}, C_{5Y}, C_{5Z}, C_3, \\ C_4, \sigma, tag_n)$ is obtained. The re-encryption processing unit
44 writes the obtained re-encrypted data C_j into the
temporary data storage unit 42.

25 If verification processing in step ST51' in decryption

processing described later is omitted, the re-encrypted data

$C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4, \sigma, \text{tag}_n)$ may be set as

5 $C_j = (C_{2X}'', C_{2Y}'', C_{2Z}'', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3)$ by omitting verification data $(C_1, C_{2X}', C_{2Y}', C_{2Z}', C_{2Z1}', C_4, \sigma, \text{tag}_n)$ that is not used for decryption. In this case, processing to generate the verification data $(C_{2X}', C_{2Y}', C_{2Z}', C_{2Z1}')$ is also omitted.

10 In any case, the communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of the control unit 46 (ST43').

(5) Decryption processing

15 The decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in the temporary data storage unit 52 using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit
20 52 based on the following verification formulas (ST51').

$$\begin{aligned}
 e(C''_{2X}, u^{C_1} \cdot v) &= e(C'_{2X}, C_4), \\
 e(C''_{2Y}, u^{C_1} \cdot v) &= e(C'_{2Y}, C_4), \\
 e(C''_{2Z}, u^{C_1} \cdot v) &= e(C'_{2Z}, C_4), \\
 e(C''_{2Z1}, u^{C_1} \cdot v) &= e(C'_{2Z1}, C_4), \\
 e(C_{5Z}, C'_{2Z}) &= e(C_{5X}, C'_{2X}) \cdot e(g_2, g_{3,n})^{y_j}, \\
 e(C_{5Z}, C'_{2Z1}) &= e(C_{5Y}, C'_{2Y}) \cdot e(g, g_{4,n})^{x_j}, \\
 \hat{V}(C_1, \sigma, (\text{tag}_n, C_3, C_4)) &= 1. \quad \dots (a7)
 \end{aligned}$$

Instead of the fifth verification formula in the formula (a7), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) = e(C_{5X}, C'_{2X}) \cdot e(g_{5,n}, g)^{y_j}$$

Also, instead of the fifth verification formula in the formula (a7), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) = e(C_{5X}, C'_{2X}) \cdot e(Y_{2j}, g_{3,n}) \quad \dots (a7')$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

Also, instead of the sixth verification formula in the formula (a7), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(g_{3,n}, g_1)^{x_j}$$

Also, instead of the sixth verification formula in the formula (a7), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_{4,n})$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

5 The decryption processing unit 54 of the decryption device 50 may select, based on the identification information tag_n of A_n contained in the re-encrypted data C_j transmitted in step ST43', A_n to be used in step ST51' from $A_1, A_2, \dots, A_N$ as elements of the public parameter par .

10 The decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

15 That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi_n(x_j + \beta y_j + \theta)/z_i k}, g^{z_i r k})}{e(g^{\pi_n(x_j + \theta)/x_i s}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi_n(x_j + \beta y_j + \theta)/z_i k}, g^{\alpha z_i r k})}{e(g^{\pi_n(\beta y_j + \theta)/y_i t}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \end{aligned}$$

$$\begin{aligned} &= C_3 / \left\{ \left(\frac{e(g^{\pi_n(x_j + \beta y_j + \theta)}, g^r)}{e(g^{\pi_n(x_j + \theta)}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi_n(x_j + \beta y_j + \theta)}, g^{\alpha r})}{e(g^{\pi_n(\beta y_j + \theta)}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\ &= m \cdot e(g_{4,n} g_{5,n}, g)^r / (e(g, g)^{\pi_n \beta r} \cdot e(g, g)^{\pi_n \alpha r}) = m. \end{aligned}$$

[Sixth modification]

In the fifth modification, an example in which
5 ciphertext data is generated by the encryption device 20,
the ciphertext data is re-encrypted by the re-encryption
device 40 to generate re-encrypted data, and the re-
encrypted data is decrypted by the decryption device 50 is
cited, but the first embodiment may also be changed to a
10 form in which the ciphertext data is decrypted without
being re-encrypted. In this case, only the key setup
processing, encryption processing, and decryption
processing need to be performed. The key setup processing
in the present modification is the same as the key setup
15 processing in the fifth modification. Hereinafter, the
encryption processing and the decryption processing in the
present modification will be described.

The difference between the encryption processing in
the present modification and the encryption processing in
20 the fifth modification is only in the last step. To
provide a description by making use of the above symbols,
the identification information of the decryption device 50
is assumed to be i for the sake of convenience. In this
case, the communication unit 22 of the encryption device 20
25 transmits the ciphertext data C_i in the temporary data

storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies, like in step ST51' in the first modification, the ciphertext data C_j generated by the encryption device 20 (ST51').

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_{4,n}g_{5,n}, C_{2X})^{\frac{1}{x_i}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_{4,n}g_{5,n}, g)^r / e(g_{4,n}g_{5,n}, C_{2X})^{\frac{1}{x_i}} \\ &= m \cdot e(g_{4,n}g_{5,n}, g)^r / e(g_{4,n}g_{5,n}, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_{4,n}g_{5,n}, g)^r / e(g_{4,n}g_{5,n}, g)^r = m. \end{aligned}$$

The decryption processing unit 54 of the decryption device 50 may select, based on the identification information tag_n of A_n contained in the ciphertext data C_i transmitted in step ST24', A_n to be used in step ST52' from $A_1, A_2, \dots, A_N$ as elements of the public parameter par .

[Seventh modification]

In addition to the sixth modification, as will be

described below, the first embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption

5 processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the fifth modification respectively.

Hereinafter, the encryption processing and the decryption
10 processing in the present modification will be described. In the present modification, the identification information of the decryption device 50 is assumed to be j .

Like in step ST21 in the first embodiment, the encryption parameter generation unit 23 of the encryption
15 device 20 generates (ssk, svk) (ST21') and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

The ciphertext generation unit 24 generates one element from $A_1, A_2, \dots, A_N$ as elements of the public parameter par . It is assumed below that A_n is selected.

20 The encryption parameter generation unit 23 also generates and sends out five random numbers $r, s, t, k, \theta \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2x}', C_{2x}'', C_{2y}', C_{2y}'', C_{2z}', C_{2z}'', C_{2z1}', C_{2z1}'', C_{5x},$
25 C_{5y}, C_{5z}, C_3, C_4 below from a message $m \in G_T$ as plaintext data using these random numbers r, s, t, k, θ , the public key pk_j of the decryption device 50, and the public parameter (ST22').

$$\begin{aligned} C'_{2X} &= Y_{2j}^s, C''_{2X} = Y_{2j}^{rs}, C'_{2Y} = X_j^t, C''_{2Y} = X_j^{rt}, \\ C'_{2Z} &= Y_{2j}^k, C''_{2Z} = Y_{2j}^{rk}, C'_{2Z1} = X_j^k, C''_{2Z1} = X_j^{rk}, \\ C_3 &= e(g_{4,n}g_{5,n}, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r, \\ C_{5X} &= (g_{4,n} \cdot g_{3,n}^\theta)^{\frac{1}{s}}, C_{5Y} = g_{3,n}^{\frac{\theta+1}{t}}, C_{5Z} = (g_{4,n} \cdot g_{3,n}^{\theta+1})^{\frac{1}{k}} \end{aligned}$$

After step ST22' is completed, like in step ST23 in the fifth modification, the ciphertext generation unit 24 generates a one-time signature σ (ST23').

Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4, \sigma, \text{tag}_n)$ including all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_j generated by the encryption device 20 in the same manner as in step ST51' in the fifth modification (ST51').

The decryption processing unit 54 of the decryption device 50 may select, based on the identification information tag_n of A_n contained in the ciphertext data C_1 transmitted in step ST24', A_n to be used in step ST51' from $A_1, A_2, \dots, A_N$ as elements of the public parameter par .

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi_n(1+\alpha+\theta)/k}, g^{\beta y_j r k})}{e(g^{\pi_n(\alpha+\theta)/s}, g^{\beta y_j r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi_n(1+\alpha+\theta)/k}, g^{x_j r k})}{e(g^{\pi_n(\theta+1)/t}, g^{x_j r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi_n(1+\alpha+\theta)}, g^{\beta r})}{e(g^{\pi_n(\alpha+\theta)}, g^{\beta r})} \right) \cdot \left(\frac{e(g^{\pi_n(1+\alpha+\theta)}, g^r)}{e(g^{\pi_n(\theta+1)}, g^r)} \right) \right\} \\ &= m \cdot e(g_{4,n} g_{5,n}, g)^r / (e(g, g)^{\pi_n \beta r} \cdot e(g, g)^{\pi_n \alpha r}) = m. \end{aligned}$$

(Second embodiment)

In the present embodiment, an example of updating a re-encryption key in the first embodiment at certain intervals will be described. A time parameter L is used as

a parameter representing a period. The time parameter is used in, of key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing, the encryption processing, the re-encryption key generation processing, and the re-encryption processing. The time parameter is used such that if, for example, encryption processing is performed in a certain period t_1 , the encryption processing described later is performed by setting $L=t_1$ and if encryption processing is performed in the next period t_2 , the encryption processing described later is performed by setting $L=t_2$. That is, encryption processing, re-encryption key generation processing, and re-encryption processing are performed by periodically changing the time parameter. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

A public parameter generation unit 13 of a key generator 10 generates a public parameter par in the same manner as in the first embodiment or acquires the parameter from outside (ST1). If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects elements x_i, y_i, z_i, w_i of Z_p^* from Z_p^*

and sets a private key $sk_i=(x_i, y_i, z_i, w_i)$ of the re-
encryption key generator 30. Then, the key generation unit
14 generates a public key $pk_i=(X_i, Y_{1i}, Y_{2i}, Z_i, Z_{1i}, W_i)$ of the
re-encryption key generator 30 using the private key sk_i
5 and the public parameter par.

$$X_i = g^{x_i}, \quad Y_{1i} = g_1^{y_i}, \quad Y_{2i} = g_2^{y_i}, \\ Z_i = g^{z_i}, \quad Z_{1i} = g_1^{z_i}, \quad W_i = g^{w_i}$$

When verification processing using a formula (a6')
described later is not performed, data Y_{2i} may be omitted.
That is, the public key pk_i only needs to include $X_i, Y_{1i},$
10 Z_i, Z_{1i}, W_i .

Further, the key generation unit 14 generates a re-
encryption key generation key $rk_i=(X_{3i}, Y_{5i})$ using the
private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{x_i}, \quad Y_{5i} = g_5^{y_i}$$

15 Similarly, if the identification information of a
decryption device 50 is j , the key generation unit 14
randomly selects elements x_j, y_j, z_j, w_j of Z_p^* from Z_p^* and
sets a private key $sk_j=(x_j, y_j, z_j, w_j)$ of the decryption
device 50. Then, the key generation unit 14 generates a
20 public key $pk_j=(X_j, Y_{1j}, Y_{2j}, Z_j, Z_{1j}, W_j)$ of the decryption device
50 using the private key sk_j and the public parameter par.

$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Y_{2j} = g_2^{y_j}, \\ Z_j = g^{z_j}, \quad Z_{1j} = g_1^{z_j}, \quad W_j = g^{w_j}$$

If, similarly as described above, the verification

processing using a formula (a6') described later is not performed, data Y_{2j} may be omitted. That is, the public key pk_j only needs to include $X_j, Y_{1j}, Z_j, Z_{1j}, W_j$.

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j=(X_{3j}, Y_{5j})$ using the private key sk_j and the public parameter par (ST6).

$$X_{3j}=g_3^{X_j}, \quad Y_{5j}=g_5^{Y_j}$$

(2) Encryption processing

Like in step ST21 in the first embodiment, an encryption parameter generation unit 23 of an encryption device 20 generates (ssk, svk) (ST21) and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

The encryption parameter generation unit 23 also generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random number r , the public key pk_i of the re-encryption key generator 30, the public parameter par , and the time parameter L (ST22).

$$C_{2X} = X_i^r, \quad C_{2Y} = Y_{1i}^r, \quad C_{2Z} = Z_i^r, \quad C_{2Z1} = Z_{1i}^r, \\ C_{2F} = F_i(L)^r, \quad C_3 = e(g_4 g_5, g)^r \cdot m, \quad C_4 = (u^{svk} \cdot v)^r$$

The function $F_i(L)$ is defined as shown below.

$$F_i(L) = g^L \cdot W_i = g^{L+w_i} \quad (L \in \mathbb{Z}_p^*)$$

As a method of acquiring the time parameter by the encryption device 20 (or the re-encryption key generator 30 described later), for example, the following method may be used. Some entity (for example, a server Sv in FIG. 8, a
5 server separate from the server Sv, a user administrator, a group administrator, an administrator separate from the user administrator and the group administrator, a user A, a user B, a user B, or a user separate from the users A to C) holds a time parameter generation device (not shown) that
10 generates a time parameter and the time parameter generation device notifies the encryption device 20 (or the re-encryption key generator 30) of a time parameter. Alternatively, a function in which the input is a time and the output is a time parameter may be defined to generate a
15 time parameter using the time and the function. In this case, the encryption device 20 (or the re-encryption key generator 30) may use an output value of the function using, as input, information based on the time when encryption
20 processing (re-encryption key generation processing) is performed (for example, a value obtained by converting the date when encryption processing (re-encryption key generation processing) is performed into a number) as a time parameter L, which obviates the need to acquire a time parameter from another device. Alternatively, the key
25 generator 10, the encryption device 20, the re-encryption key generator 30, a re-encryption device 40, or the decryption device 50 may generate a time parameter to

notify devices other than the relevant device (that has generated the time parameter) of the time parameter. The re-encryption key generator 30 acquires, as will be described later, a re-encryption key generation key of the decryption device 50 from the decryption device 50 and when acquiring the re-encryption key generation key, may also acquire a time parameter from the decryption device 50. In this case, the decryption device 50 may acquire a time parameter from one of the above methods. This also applies to embodiments and modifications described later.

The time parameter does not necessarily have to be a parameter related to a time such as a time or date and may be, for example, a number obtained by converting a character string or a numerical value such as a random number. Similarly, the input into the above function may be a number obtained by converting a character string or a numerical value such as a random number. Thus, the value of a time parameter may be unrelated to time. This also applies to embodiments and modifications described later.

After step ST22 is completed, the ciphertext generation unit 24 generates a one-time signature σ as shown in the following formula for the time parameter L and the encrypted data C_3, C_4 based on a signature generation function $\hat{S}$ of the public parameter par and the signature key ssk generated in step ST21 (ST23).

$$\sigma = \hat{S}(ssk, (L, C_3, C_4))$$

Incidentally, the ciphertext generation unit 24 may

generate the one-time signature σ for the encrypted data C_3 , C_4 as data to be signed without including the time parameter L . This also applies to embodiments and modifications described later.

5 Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_i = (L, C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4, \sigma)$ including the time parameter L , all the above encrypted data C_1 to C_4 , and the one-time signature σ and writes the obtained ciphertext data into a temporary data
10 storage unit 21.

 A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to the re-encryption device 40 under the control of a control unit 25 (ST24).

15 (3) Re-encryption key generation processing

 Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) and
20 generated by using L as the time parameter value is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50
25 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator

30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35 (ST31).

5 A random number generation unit 36 generates and sends out three random numbers $\theta, \delta x, \delta y \in \mathbb{Z}_p^*$ to a re-encryption key generation unit 34.

 The re-encryption key generation unit 34 generates a re-encryption key R_{ijL} as shown in the following formulas using the random numbers $\theta, \delta x, \delta y$, the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, the public parameter par , and the time parameter L (ST32).

15 $R_{ijL} = (R_{ijL1}, R_{ijL2}, R_{ijL3}, R_{ijL4}, R_{ijL5})$

$$R_{ijL1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} \cdot g^{(L+w_i)\delta x} = g^{\frac{\pi(x_j+\theta)}{x_i} + (L+w_i)\delta x},$$

$$R_{ijL2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} \cdot g^{(L+w_i)\delta y} = g^{\frac{\pi(\beta y_j+\theta)}{y_i} + (L+w_i)\delta y},$$

$$R_{ijL3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j+\beta y_j+\theta)}{z_i}},$$

$$R_{ijL4} = g^{x_i \delta x}, R_{ijL5} = g_1^{y_i \delta y} = g^{\alpha y_i \delta y}$$

In the above formulas, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

(4) Re-encryption processing

20 A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a

temporary data storage unit 42 using the public parameter
par, the time parameter L contained in the ciphertext data
C_i in the temporary data storage unit 42, and the public
key pk_i of the re-encryption key generator 30 based on the
5 following verification formula (ST41).

$$\begin{aligned} e(C_{2X}, u^{C_1} \cdot v) &= e(X_i, C_4), \\ e(C_{2Y}, u^{C_1} \cdot v) &= e(Y_{1i}, C_4), \\ e(C_{2Z}, u^{C_1} \cdot v) &= e(Z_i, C_4), \\ e(C_{2Z1}, u^{C_1} \cdot v) &= e(Z_{1i}, C_4), \\ e(C_{2F}, u^{C_1} \cdot v) &= e(F_i(L), C_4), \\ \hat{V}(C_1, \sigma, (L, C_3, C_4)) &= 1. \end{aligned} \quad \dots (a5)$$

If the time parameter L is not included as data to be
signed by the one-time signature σ , the sixth verification
formula in the formula (a5) conducts verification of the
10 one-time signature σ by restricting data to be signed to
the encrypted data C₃, C₄. This also applies to
verification processing of the one-time signature σ
described later.

The verification is successful if the six verification
15 formulas in the formula (a5) all hold and fails if any one
of the six formulas does not hold.

If the verification is successful (or the verification
processing in step ST41 is skipped), a re-encryption

parameter generation unit 45 generates and sends out four random numbers $s, t, k, h \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

5 The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' , C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{2F}' , C_{2F}'' , C_{5X} , C_{5Y} , C_{5Z} , C_{5FX} , C_{5FY} using these random numbers s, t, k, h , the ciphertext data C_i stored in the temporary data storage unit 42, the re-encryption key R_{ijL} , the public key pk_i of
10 the re-encryption key generator 30, and the public parameter par (ST42).

$$\begin{aligned} C'_{2X} &= X_i^s, C''_{2X} = C_{2X}^s = X_i^{rs}, C'_{2Y} = Y_{1i}^t, C''_{2Y} = C_{2Y}^t = Y_{1i}^{rt}, \\ C'_{2Z} &= Z_i^k, C''_{2Z} = C_{2Z}^k = Z_i^{rk}, C'_{2Z1} = Z_{1i}^k, C''_{2Z1} = C_{2Z1}^k = Z_{1i}^{rk}, \\ C'_{2F} &= F_i(L)^h, C''_{2F} = C_{2F}^h = F_i(L)^{rh}, \\ C_{5X} &= R_{ijL1}^{\frac{1}{s}}, C_{5Y} = R_{ijL2}^{\frac{1}{t}}, C_{5Z} = R_{ijL3}^{\frac{1}{k}}, \\ C_{5FX} &= R_{ijL4}^{\frac{1}{h}}, C_{5FY} = R_{ijL5}^{\frac{1}{h}} \end{aligned}$$

15 After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data $C_j = (L, C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, C_3, C_4, \sigma)$ by replacing the encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}$ in the ciphertext data C_i with all the
20 above re-encrypted data C_{2X}' to C_{5FY} and writes the obtained ciphertext data C_j into the temporary data storage unit 42.

A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the

decryption device 50 under the control of a control unit 46 (ST43).

(5) Decryption processing

5 A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51 and the public parameter par in the temporary data storage unit 52 based on the following verification formulas (ST51).

$$e(C''_{2X}, u^{C_1} \cdot v) = e(C'_{2X}, C_4),$$

$$e(C''_{2Y}, u^{C_1} \cdot v) = e(C'_{2Y}, C_4),$$

$$e(C''_{2Z}, u^{C_1} \cdot v) = e(C'_{2Z}, C_4),$$

$$e(C''_{2Z1}, u^{C_1} \cdot v) = e(C'_{2Z1}, C_4),$$

$$e(C''_{2F}, u^{C_1} \cdot v) = e(C'_{2F}, C_4),$$

$$e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) = e(C_{5X}, C'_{2X}) \cdot e(g_2, g_3)^{y_j},$$

$$e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) = e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j},$$

$$\hat{V}(C_1, \sigma, (L, C_3, C_4)) = 1.$$

10 ... (a6)

Instead of the sixth verification formula in the formula (a6), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) = e(C_{5X}, C'_{2X}) \cdot e(g_5, g)^{y_j}.$$

15 Also, instead of the sixth verification formula in the formula (a6), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) = e(C_{5X}, C'_{2X}) \cdot e(Y_{2j}, g_3). \dots (a6')$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

5 Also, instead of the seventh verification formula in the formula (a6), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) = e(C_{5Y}, C'_{2Y}) \cdot e(g_3, g_1)^{x_j}.$$

10 Also, instead of the seventh verification formula in the formula (a6), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4).$$

15 In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

The verification is successful if the eight verification formulas in the formula (a6) all hold and fails if any one of the eight formulas does not hold.

20 If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit
25 52 to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z}) \cdot e(C_{5FX}, C''_{2F})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1}) \cdot e(C_{5FY}, C''_{2F})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z}) \cdot e(C_{5FX}, C''_{2F})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1}) \cdot e(C_{5FY}, C''_{2F})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\frac{\pi(x_j+\beta y_j+\theta)}{z_i k}}, g^{z_i r k}) \cdot e(g^{\frac{x_i \delta_x}{h}}, g^{(L+w_i) r h})}{e(g^{\frac{(\pi(x_j+\theta)/x_i)+(L+w_i)\delta_x}{s}}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\pi(x_j+\beta y_j+\theta)}{z_i k}}, g^{\alpha z_i r k}) \cdot e(g^{\frac{\alpha y_i \delta_y}{h}}, g^{(L+w_i) r h})}{e(g^{\frac{(\pi(\beta y_j+\theta)/y_i)+(L+w_i)\delta_y}{t}}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi(x_j+\beta y_j+\theta)}, g^r) \cdot e(g^{x_i \delta_x}, g^{(L+w_i) r})}{e(g^{\pi(x_j+\theta)+(L+w_i)x_i \delta_x}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi(x_j+\beta y_j+\theta)}, g^{\alpha r}) \cdot e(g^{\alpha y_i \delta_y}, g^{(L+w_i) r})}{e(g^{\pi(\beta y_j+\theta)+(L+w_i)y_i \delta_y}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\ &= m \cdot e(g_4 g_5, g)^r / (e(g, g)^{\pi \beta r} \cdot e(g, g)^{\pi \alpha r}) = m. \end{aligned}$$

In the present embodiment, the order of processing may be changed when appropriate. For example, the order of the decryption processing and the ciphertext verification processing may be changed. Similarly, the order of the re-encryption key generation processing may be changed to before the encryption processing.

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ijL} based on the random numbers θ , δx , δy ,

the re-encryption key generation key rk_j , and the time parameter L , in addition to the effects of the first embodiment, whether to transfer the decryption right can be decided in each period so that finer access control can be exercised. Accordingly, even if the decryption right of ciphertext for the transfer source (in the example of FIG. 8, the group administrator) is once transferred to the transfer destination (in the example of FIG. 8, for example, the user A) in a certain period, it is possible not to grant the decryption right of the ciphertext for the transfer source to the relevant transfer destination, that is, the decryption right of the transfer destination (of the ciphertext for the relevant transfer source) can be invalidated in the next period and therefore, a file sharing system whose convenience is further improved can be provided.

[Eighth modification]

In the second embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is cited, but the second embodiment may be changed to a form in which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present

modification is the same as the key setup processing in the second embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

5 The difference between the encryption processing in the present modification and the encryption processing in the second embodiment is in the last step only. To provide a description by making use of the above symbols, the identification information of the decryption device 50 is
10 assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

15 The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formulas (ST51').

$$\begin{aligned} e(C_{2X}, u^{C_1} \cdot v) &= e(g, C_4)^{x_i}, \\ e(C_{2Y}, u^{C_1} \cdot v) &= e(g_1, C_4)^{y_i}, \\ e(C_{2Z}, u^{C_1} \cdot v) &= e(g, C_4)^{z_i}, \\ e(C_{2Z1}, u^{C_1} \cdot v) &= e(g_1, C_4)^{z_i}, \\ e(C_{2F}, u^{C_1} \cdot v) &= e(g, C_4)^{L+w_i}, \\ \hat{V}(C_1, \sigma, (L, C_3, C_4)) &= 1. \end{aligned}$$

20 Incidentally, the decryption device 50 may verify the

ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

5 The verification is successful if the six verification formulas shown above all hold and fails if any one of the six formulas does not hold.

10 If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_4g_5, C_{2X})^{\frac{1}{x_i}}$$

15 That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_4g_5, g)^r / e(g_4g_5, C_{2X})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_5, g)^r / e(g_4g_5, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_5, g)^r / e(g_4g_5, g)^r = m. \end{aligned}$$

[Ninth modification]

20 In addition to the eighth modification, as will be described below, the second embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption

processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the second embodiment respectively.

5 Hereinafter, the encryption processing and the decryption processing in the present modification will be described. In the present modification, the identification information of the decryption device 50 is assumed to be j .

10 Like in step ST21 in the first embodiment, the encryption parameter generation unit 23 of the encryption device 20 generates (ssk, svk) (ST21') and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

15 The encryption parameter generation unit 23 also generates and sends out eight random numbers $r, s, t, k, h, \theta, \delta x, \delta y \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

20 The ciphertext generation unit 24 generates encrypted data $C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using these random numbers $r, s, t, k, h, \theta, \delta x, \delta y$, the public key pk_j of the decryption device 50, the public parameter, and the time parameter L (ST22').

$$\begin{aligned}
C'_{2X} &= Y_{2j}^s, \quad C''_{2X} = Y_{2j}^{rs}, \quad C'_{2Y} = X_j^t, \quad C''_{2Y} = X_j^{rt}, \\
C'_{2Z} &= Y_{2j}^k, \quad C''_{2Z} = Y_{2j}^{rk}, \quad C'_{2Z1} = X_j^k, \quad C''_{2Z1} = X_j^{rk}, \\
C'_{2F} &= F_j(L)^h, \quad C''_{2F} = F_j(L)^{rh}, \\
C_3 &= e(g_4 g_5, g)^r \cdot m, \quad C_4 = (u^{svk} \cdot v)^r, \\
C_{5X} &= (g_4 \cdot g_3^\theta \cdot F_j(L)^{\delta_y})^{\frac{1}{s}} = g^{\frac{\pi(\alpha+\theta)+(L+w_j)\delta_y}{s}}, \\
C_{5Y} &= (g_3^{1+\theta} \cdot F_j(L)^{\delta_x})^{\frac{1}{t}} = g^{\frac{\pi(1+\theta)+(L+w_j)\delta_x}{t}}, \\
C_{5Z} &= (g_4 \cdot g_3^{1+\theta})^{\frac{1}{k}} = g^{\frac{\pi(\alpha+1+\theta)}{k}}, \\
C_{5FX} &= (Y_{2j})^{\frac{\delta_y}{h}}, \quad C_{5FY} = (X_j)^{\frac{\delta_x}{h}}
\end{aligned}$$

After step ST22' is completed, like in step ST23, the ciphertext generation unit 24 generates a one-time signature σ .

Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_i = (L, C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, C_3, C_4, \sigma)$ including the time parameter L , all the above encrypted data C_1 to C_4 and, the one-time signature σ and writes the obtained ciphertext data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_j generated by the encryption device 20 in the same manner as in step ST51 (ST51'). If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j and the public parameter par to obtain the message m (ST52').

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\frac{\pi(\alpha+1+\theta)}{k}}, g^{\beta y_j r k}) \cdot e(g^{\frac{\beta y_j \delta_y}{h}}, g^{(L+w_j) r h})}{e(g^{\frac{\pi(\alpha+\theta)+(L+w_j)\delta_y}{s}}, g^{\beta y_j r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\pi(\alpha+1+\theta)}{k}}, g^{x_j r k}) \cdot e(g^{\frac{x_j \delta_x}{h}}, g^{(L+w_j) r h})}{e(g^{\frac{\pi(1+\theta)+(L+w_j)\delta_x}{t}}, g^{x_j r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi(\alpha+1+\theta)}, g^{\beta r}) \cdot e(g^{\beta \delta_y}, g^{(L+w_j) r})}{e(g^{\pi(\alpha+\theta)+(L+w_j)\delta_y}, g^{\beta r})} \right) \cdot \left(\frac{e(g^{\pi(\alpha+1+\theta)}, g^r) \cdot e(g^{\delta_x}, g^{(L+w_j) r})}{e(g^{\pi(1+\theta)+(L+w_j)\delta_x}, g^r)} \right) \right\} \\ &= m \cdot e(g_4 g_5, g)^r / (e(g, g)^{\pi \beta r} \cdot e(g, g)^{\pi \alpha r}) = m. \end{aligned}$$

(Third embodiment)

In the present embodiment, an example of omitting the encryption function shown in the second modification in the

first embodiment will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

First, a public parameter generation unit 13 of a key generator 10 generates a public parameter

$$\text{par} = \{\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_2, g_3, g_4, u, v, \text{Sig}\}$$

based on a security parameter λ stored in a key generation parameter storage unit 11 in advance or acquires the public parameter from outside (ST1).

Here, $g_3 = g^n$ and $g_4 = g_1^n$.

If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects elements x_i, y_i, z_i of Z_p^* from Z_p^* and sets a private key $sk_i = (x_i, y_i, z_i)$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $pk_i = (X_i, Y_{1i}, Z_i, Z_{1i})$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par .

$$X_i = g^{x_i}, Y_{1i} = g_1^{y_i}, Z_i = g^{z_i}, Z_{1i} = g_1^{z_i}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i = (X_{3i}, Y_{2i})$ using the

private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{x_i}, \quad Y_{2i} = g_2^{y_i}$$

Similarly, if the identification information of a decryption device 50 is j , the key generation unit 14

5 randomly selects elements x_j, y_j, z_j of Z_p^* from Z_p^* and sets a private key $sk_j = (x_j, y_j, z_j)$ of the decryption device 50.

Then, the key generation unit 14 generates a public key $pk_j = (X_j, Y_{1j}, Z_j, Z_{1j})$ of the decryption device 50 using the private key sk_j and the public parameter par .

10
$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Z_j = g^{z_j}, \quad Z_{1j} = g_1^{z_j}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j = (X_{3j}, Y_{2j})$ using the private key sk_j and the public parameter par (ST6).

$$X_{3j} = g_3^{x_j}, \quad Y_{2j} = g_2^{y_j}$$

15 (2) Encryption processing

Like in step ST21 in the first embodiment, an encryption parameter generation unit 23 of an encryption device 20 generates (ssk, svk) (ST21) and sets the verification key svk to the encrypted data C_1 ($C_1 = svk$).

20 The encryption parameter generation unit 23 also generates and sends out a random number $r \in Z_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2x}, C_{2y}, C_{2z}, C_{2z1}, C_3, C_4$ below from a message $m \in G_T$ as
25 plaintext data using the random number r , the public key

pk_i of the re-encryption key generator 30, and the public parameter par (ST22).

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_3 = e(g_4 g_2, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

After step ST22 is completed, like in step ST23 in the first embodiment, the ciphertext generation unit 24 generates a one-time signature σ for the encrypted data C_3, C_4 based on a signature generation function $\hat{S}$ of the public parameter par and the signature key ssk generated in step ST21 (ST23).

Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_1 = (C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, \sigma)$ including all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext data into a temporary data storage unit 21.

A communication unit 22 transmits the ciphertext data C_1 in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the

ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35 (ST31).

A random number generation unit 36 generates and sends out a random number $\theta \in \mathbb{Z}_p^*$ to a re-encryption key generation unit 34.

The re-encryption key generation unit 34 generates a re-encryption key R_{ij} as shown in the following formulas using the random number θ , the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, and the public parameter par (ST32).

$$R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$$

$$R_{ij1} = (X_{3j} \cdot g^\theta)^{1/x_i} = g^{\frac{\pi x_j + \theta}{x_i}},$$

$$R_{ij2} = (Y_{2j} \cdot g^\theta)^{1/y_i} = g^{\frac{\beta y_j + \theta}{y_i}},$$

$$R_{ij3} = (X_{3j} \cdot Y_{2j} \cdot g^\theta)^{1/z_i} = g^{\frac{\pi x_j + \beta y_j + \theta}{z_i}}.$$

In the above formulas, g_2 is rewritten and denoted by

g^β .

(4) Re-encryption processing

A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par and the public key pk_i of the re-encryption key generator 30 based on the following verification formula (ST41).

$$\begin{aligned}e(C_{2X}, u^{C_1} \cdot v) &= e(X_i, C_4), \\e(C_{2Y}, u^{C_1} \cdot v) &= e(Y_{1i}, C_4), \\e(C_{2Z}, u^{C_1} \cdot v) &= e(Z_i, C_4), \\e(C_{2Z1}, u^{C_1} \cdot v) &= e(Z_{1i}, C_4), \\ \hat{V}(C_1, \sigma, (C_3, C_4)) &= 1.\end{aligned}$$

The verification is successful if the five verification formulas shown above all hold and fails if any one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out three random numbers $s, t, k \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data $C_{2X'}$, $C_{2X''}$, $C_{2Y'}$, $C_{2Y''}$, $C_{2Z'}$, $C_{2Z''}$, $C_{2Z1'}$, $C_{2Z1''}$, C_{5X} , C_{5Y} , C_{5Z} using these random numbers s, t, k , the ciphertext data C_i

stored in the temporary data storage unit 42, the re-encryption key R_{ij} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$\begin{aligned} C'_{2X} &= X_i^s, C''_{2X} = C_{2X}^s = X_i^{rs}, C'_{2Y} = Y_{1i}^t, C''_{2Y} = C_{2Y}^t = Y_{1i}^{rt}, \\ C'_{2Z} &= Z_i^k, C''_{2Z} = C_{2Z}^k = Z_i^{rk}, C'_{2Z1} = Z_{1i}^k, C''_{2Z1} = C_{2Z1}^k = Z_{1i}^{rk}, \\ C_{5X} &= R_{ij1}^{\frac{1}{s}}, C_{5Y} = R_{ij2}^{\frac{1}{t}}, C_{5Z} = R_{ij3}^{\frac{1}{k}} \end{aligned}$$

5

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data

$C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4, \sigma)$ by replacing the encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}$ in the ciphertext data C_i with all the above re-encrypted data C_{2X}' to C_{5Z} and writes the obtained re-encrypted data C_j into the temporary data storage unit 42.

10

A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46 (ST43).

15

(5) Decryption processing

A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51 and the public parameter par in the temporary data storage unit 52 based on the following verification formulas (ST51).

20

$$\begin{aligned}
e(C''_{2X}, u^{C_1} \cdot v) &= e(C'_{2X}, C_4), \\
e(C''_{2Y}, u^{C_1} \cdot v) &= e(C'_{2Y}, C_4), \\
e(C''_{2Z}, u^{C_1} \cdot v) &= e(C'_{2Z}, C_4), \\
e(C''_{2Z1}, u^{C_1} \cdot v) &= e(C'_{2Z1}, C_4), \\
e(C_{5Z}, C'_{2Z}) &= e(C_{5X}, C'_{2X}) \cdot e(g_2, g)^{y_j}, \\
e(C_{5Z}, C'_{2Z1}) &= e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j}, \\
\hat{V}(C_1, \sigma, (C_3, C_4)) &= 1. \quad \dots (b1)
\end{aligned}$$

Instead of the sixth verification formula in the formula (b1), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4).$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

The verification is successful if the seven verification formulas in the formula (b1) all hold and fails if any one of the seven formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{(\pi x_j + \beta y_j + \theta)/z_i k}, g^{z_i r k})}{e(g^{(\pi x_j + \theta)/x_i s}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{(\pi x_j + \beta y_j + \theta)/z_i k}, g^{\alpha z_i r k})}{e(g^{(\beta y_j + \theta)/y_i t}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^r)}{e(g^{\pi x_j + \theta}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^{\alpha r})}{e(g^{\beta y_j + \theta}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\ &= m \cdot e(g_4 g_2, g)^r / (e(g, g)^{\beta r} \cdot e(g, g)^{\pi \alpha r}) = m. \end{aligned}$$

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ij} based on the random number θ and the re-encryption key generation key rk_j , effects similar to those in the first embodiment can be obtained.

Also according to the present embodiment, as described above, due to the configuration of omitting g_5 in the first embodiment, the data size of the public parameter and public key can be reduced when compared with the embodiment shown in the first embodiment. Also according to the

present embodiment, the amount of computations needed for

key setup processing can be reduced when compared with the embodiment shown in the first embodiment.

[Tenth modification]

In the third embodiment, an example in which
5 ciphertext data is generated by the encryption device 20,
the ciphertext data is re-encrypted by the re-encryption
device 40 to generate re-encrypted data, and the re-
encrypted data is decrypted by the decryption device 50 is
cited, but the third embodiment may be changed to a form in
10 which the ciphertext data is decrypted without being re-
encrypted. In this case, only the key setup processing,
encryption processing, and decryption processing need to be
performed. The key setup processing in the present
modification is the same as the key setup processing in the
15 third embodiment. Hereinafter, the encryption processing
and the decryption processing in the present modification
will be described.

The difference between the encryption processing in
the present modification and the encryption processing in
20 the third embodiment is in the last step only. To provide
a description by making use of the above symbols, the
identification information of the decryption device 50 is
assumed to be i for the sake of convenience. In this case,
the communication unit 22 of the encryption device 20
25 transmits the ciphertext data C_i in the temporary data
storage unit 21 to the decryption device 50 under the
control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formulas (ST51').

$$\begin{aligned}e(C_{2X}, u^{C_1} \cdot v) &= e(g, C_4)^{x_i}, \\e(C_{2Y}, u^{C_1} \cdot v) &= e(g_1, C_4)^{y_i}, \\e(C_{2Z}, u^{C_1} \cdot v) &= e(g, C_4)^{z_i}, \\e(C_{2Z1}, u^{C_1} \cdot v) &= e(g_1, C_4)^{z_i}, \\ \hat{V}(C_1, \sigma, (C_3, C_4)) &= 1.\end{aligned}$$

5

... (b2)

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

10

The verification is successful if the five verification formulas in the formula (b2) all hold and fails if any one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

15

$$m = C_3 / e(g_4 g_2, C_{2X})^{\frac{1}{x_i}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_4g_2, g)^r / e(g_4g_2, C_2X)^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_2, g)^r / e(g_4g_2, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_2, g)^r / e(g_4g_2, g)^r = m. \end{aligned}$$

5

(Fourth embodiment)

In the present embodiment, an example of updating a re-encryption key in the third embodiment at certain intervals will be described. Hereinafter, key setup
10 processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment or second embodiment may or may not be described
15 by using the same reference signs.

(1) Key setup processing

A public parameter generation unit 13 of a key generator 10 generates a public parameter par in the same manner as in the third embodiment or acquires the parameter
20 from outside (ST1). If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects elements x_i, y_i, z_i, w_i of Z_p^* from Z_p^* and sets a private key $sk_i = (x_i, y_i, z_i, w_i)$ of the re-encryption key generator 30. Then, the key generation unit

14 generates a public key $pk_i=(X_i,Y_{1i},Z_i,Z_{1i},W_i)$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par .

$$X_i = g^{x_i}, \quad Y_{1i} = g_1^{y_i}, \quad Z_i = g^{z_i}, \\ Z_{1i} = g_1^{z_i}, \quad W_i = g^{w_i}$$

5 Further, the key generation unit 14 generates a re-encryption key generation key $rk_i=(X_{3i},Y_{2i})$ using the private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{x_i}, \quad Y_{2i} = g_2^{y_i}$$

10 Similarly, if the identification information of a decryption device 50 is j , the key generation unit 14 randomly selects elements x_j, y_j, z_j, w_j of Z_p^* from Z_p^* and sets a private key $sk_j=(x_j,y_j,z_j,w_j)$ of the decryption device 50. Then, the key generation unit 14 generates a public key $pk_j=(X_j,Y_{1j},Z_j,Z_{1j},W_j)$ of the decryption device 50 using the private key sk_j and the public parameter par .

$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Z_j = g^{z_j}, \\ Z_{1j} = g_1^{z_j}, \quad W_j = g^{w_j}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j=(X_{3j},Y_{2j})$ using the private key sk_j and the public parameter par (ST6).

$$X_{3j} = g_3^{x_j}, \quad Y_{2j} = g_2^{y_j}$$

20

(2) Encryption processing

Like in step ST21 in the first embodiment, an encryption parameter generation unit 23 of an encryption device 20 generates (ssk, svk) (ST21) and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

5 The encryption parameter generation unit 23 also generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

10 The ciphertext generation unit 24 generates encrypted data C_{2X} , C_{2Y} , C_{2Z} , C_{2Z1} , C_{2F} , C_3 , C_4 below from a message $m \in G_T$ as plaintext data using the random number r , the public key pk_i of the re-encryption key generator 30, the public parameter par , and a time parameter L (ST22).

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_{2F} = F_i(L)^r, C_3 = e(g_4 g_2, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

The function $F_i(L)$ is defined as shown below.

15
$$F_i(L) = g^L \cdot W_i = g^{L+w_i} \quad (L \in \mathbb{Z}_p^*)$$

20 After step ST22 is completed, the ciphertext generation unit 24 generates a one-time signature σ as shown in the following formula for the time parameter L and the encrypted data C_3 , C_4 based on a signature generation function $\hat{S}$ of the public parameter par and the signature key ssk generated in step ST21 (ST23).

$$\sigma = \hat{S}(ssk, (L, C_3, C_4))$$

Thereafter, the ciphertext generation unit 24

generates ciphertext data $C_i = (L, C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4, \sigma)$ including the time parameter L , all the above encrypted data C_1 to C_4 , and the one-time signature σ and writes the obtained ciphertext data into a temporary data storage unit 21.

A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) and generated by using L as the time parameter value is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35 (ST31).

A random number generation unit 36 generates and sends out three random numbers $\theta, \delta x, \delta y \in \mathbb{Z}_p^*$ to a re-encryption key

generation unit 34.

The re-encryption key generation unit 34 generates a re-encryption key R_{ijL} as shown in the following formulas using the random numbers θ , δx , δy , the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, the public parameter par , and the time parameter L (ST32).

$$R_{ijL} = (R_{ijL1}, R_{ijL2}, R_{ijL3}, R_{ijL4}, R_{ijL5})$$

$$R_{ijL1} = (X_{3j} \cdot g^\theta)^{1/x_i} \cdot g^{(L+w_i)\delta x} = g^{\frac{\pi x_j + \theta}{x_i} + (L+w_i)\delta x},$$

$$R_{ijL2} = (Y_{2j} \cdot g^\theta)^{1/y_i} \cdot g^{(L+w_i)\delta y} = g^{\frac{\beta y_j + \theta}{y_i} + (L+w_i)\delta y},$$

$$R_{ijL3} = (X_{3j} \cdot Y_{2j} \cdot g^\theta)^{1/z_i} = g^{\frac{\pi x_j + \beta y_j + \theta}{z_i}},$$

$$R_{ijL4} = g^{x_i \delta x}, R_{ijL5} = g_1^{y_i \delta y} = g^{\alpha y_i \delta y}$$

In the above formulas, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

(4) Re-encryption processing

A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par , the time parameter L contained in the ciphertext data C_i in the temporary data storage unit 42, and the public key pk_i of the re-encryption key generator 30 based on the following verification formulas (ST41).

$$\begin{aligned}e(C_{2X}, u^{C_1} \cdot v) &= e(X_i, C_4), \\e(C_{2Y}, u^{C_1} \cdot v) &= e(Y_{1i}, C_4), \\e(C_{2Z}, u^{C_1} \cdot v) &= e(Z_i, C_4), \\e(C_{2Z1}, u^{C_1} \cdot v) &= e(Z_{1i}, C_4), \\e(C_{2F}, u^{C_1} \cdot v) &= e(F_i(L), C_4), \\ \hat{V}(C_1, \sigma, (L, C_3, C_4)) &= 1.\end{aligned}$$

The verification is successful if the six verification formulas shown above all hold and fails if any one of the six formulas does not hold.

5 If the verification is successful (or the verification processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out four random numbers $s, t, k, h \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

10 The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' , C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{2F}' , C_{2F}'' , C_{5X} , C_{5Y} , C_{5Z} , C_{5FX} , C_{5FY} using these random numbers s, t, k, h , the ciphertext data C_i stored in the temporary data storage
15 unit 42, the re-encryption key R_{ijL} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$\begin{aligned}
 C'_{2X} &= X_i^s, C''_{2X} = C_{2X}^s = X_i^{rs}, C'_{2Y} = Y_{1i}^t, C''_{2Y} = C_{2Y}^t = Y_{1i}^{rt}, \\
 C'_{2Z} &= Z_i^k, C''_{2Z} = C_{2Z}^k = Z_i^{rk}, C'_{2Z1} = Z_{1i}^k, C''_{2Z1} = C_{2Z1}^k = Z_{1i}^{rk}, \\
 C'_{2F} &= F_i(L)^h, C''_{2F} = C_{2F}^h = F_i(L)^{rh}, \\
 C_{5X} &= R_{ijL1}^{\frac{1}{s}}, C_{5Y} = R_{ijL2}^{\frac{1}{t}}, C_{5Z} = R_{ijL3}^{\frac{1}{k}}, \\
 C_{5FX} &= R_{ijL4}^{\frac{1}{h}}, C_{5FY} = R_{ijL5}^{\frac{1}{h}}
 \end{aligned}$$

After step ST42 is completed, the re-encryption
 5 processing unit 44 generates re-encrypted data
 $C_j = (L, C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, C_3, C_4, \sigma)$ by replacing the encrypted data
 $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}$ in the ciphertext data C_i with all the
 above re-encrypted data C_{2X}' to C_{5FY} and writes the obtained
 10 ciphertext data into the temporary data storage unit 42.

A communication unit 43 transmits the re-encrypted
 data C_j in the temporary data storage unit 42 to the
 decryption device 50 under the control of a control unit 46
 (ST43).

15 (5) Decryption processing

A decryption processing unit 54 of the decryption
 device 50 verifies the re-encrypted data C_j in a temporary
 data storage unit 52 using the private key sk_j of the local
 device 50 in a private key storage unit 51 and the public
 20 parameter par in the temporary data storage unit 52 based
 on the following verification formulas (ST51).

$$\begin{aligned}
e(C_{2X}'', u^{C_1} \cdot v) &= e(C_{2X}', C_4), \\
e(C_{2Y}'', u^{C_1} \cdot v) &= e(C_{2Y}', C_4), \\
e(C_{2Z}'', u^{C_1} \cdot v) &= e(C_{2Z}', C_4), \\
e(C_{2Z1}'', u^{C_1} \cdot v) &= e(C_{2Z1}', C_4), \\
e(C_{2F}'', u^{C_1} \cdot v) &= e(C_{2F}', C_4), \\
e(C_{5Z}, C_{2Z}') \cdot e(C_{5FX}, C_{2F}') &= e(C_{5X}, C_{2X}') \cdot e(g_2, g)^{y_j}, \\
e(C_{5Z}, C_{2Z1}') \cdot e(C_{5FY}, C_{2F}') &= e(C_{5Y}, C_{2Y}') \cdot e(g, g_4)^{x_j}, \\
\hat{V}(C_1, \sigma, (L, C_3, C_4)) &= 1. \quad \dots (b3)
\end{aligned}$$

Instead of the seventh verification formula in the formula (b3), the following verification formula may be used.

5

$$e(C_{5Z}, C_{2Z1}') \cdot e(C_{5FY}, C_{2F}') = e(C_{5Y}, C_{2Y}') \cdot e(X_j, g_4).$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

10 The verification is successful if the eight verification formulas in the formula (b3) all hold and fails if any one of the eight formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

15

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e\left(g^{\frac{\pi x_j + \beta y_j + \theta}{z_i k}}, g^{z_i r k}\right) \cdot e\left(g^{\frac{x_i \delta_x}{h}}, g^{(L+w_i)rh}\right)}{e\left(g^{\frac{((\pi x_j + \theta)/x_i) + (L+w_i)\delta_x}{s}}, g^{x_i r s}\right)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e\left(g^{\frac{\pi x_j + \beta y_j + \theta}{z_i k}}, g^{\alpha z_i r k}\right) \cdot e\left(g^{\frac{\alpha y_i \delta_y}{h}}, g^{(L+w_i)rh}\right)}{e\left(g^{\frac{((\beta y_j + \theta)/y_i) + (L+w_i)\delta_y}{t}}, g^{\alpha y_i r t}\right)} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^r) \cdot e(g^{x_i \delta_x}, g^{(L+w_i)r})}{e(g^{\pi x_j + \theta + (L+w_i)x_i \delta_x}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^{\alpha r}) \cdot e(g^{\alpha y_i \delta_y}, g^{(L+w_i)r})}{e(g^{\beta y_j + \theta + (L+w_i)y_i \delta_y}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\ &= m \cdot e(g_4 g_2, g)^r / (e(g, g)^{\beta r} \cdot e(g, g)^{\pi \alpha r}) = m. \end{aligned}$$

10

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ijL} based on the random numbers θ , δ_x , δ_y , the re-encryption key generation key rk_j , and the time parameter L , in addition to the effects of the third embodiment, whether to transfer the decryption right can be decided in each period so that finer access control can be exercised. Accordingly, even if the decryption right of

15

ciphertext for the transfer source (in the example of
FIG. 8, the group administrator) is once transferred to the
transfer destination (in the example of FIG. 8, for example,
the user A) in a certain period, it is possible not to
5 grant the decryption right of the ciphertext for the
transfer source to the relevant transfer destination, that
is, the decryption right of the transfer destination (of
the ciphertext for the relevant transfer source) can be
invalidated in the next period and therefore, a file
10 sharing system the convenience of which is further improved
can be provided.

Also according to the present embodiment, as described
above, due to the configuration of omitting g_5 in the
second embodiment, the data size of the public parameter
15 and public key can be reduced when compared with the
embodiment shown in the second embodiment. Also according
to the present embodiment, the amount of computations
needed for key setup processing can be reduced when
compared with the embodiment shown in the second embodiment.

20 [Eleventh modification]

In the fourth embodiment, an example in which
ciphertext data is generated by the encryption device 20,
the ciphertext data is re-encrypted by the re-encryption
device 40 to generate re-encrypted data, and the re-
25 encrypted data is decrypted by the decryption device 50 is
cited, but the fourth embodiment may be changed to a form
in which the ciphertext data is decrypted without being re-

encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present modification is the same as the key setup processing in the fourth embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

The difference between the encryption processing in the present modification and the encryption processing in the fourth embodiment is only in the last step. To provide a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formulas (ST51').

$$\begin{aligned}
 e(C_{2X}, u^{C_1} \cdot v) &= e(g, C_4)^{x_i}, \\
 e(C_{2Y}, u^{C_1} \cdot v) &= e(g_1, C_4)^{y_i}, \\
 e(C_{2Z}, u^{C_1} \cdot v) &= e(g, C_4)^{z_i}, \\
 e(C_{2Z1}, u^{C_1} \cdot v) &= e(g_1, C_4)^{z_i}, \\
 e(C_{2F}, u^{C_1} \cdot v) &= e(g, C_4)^{L+w_i}, \\
 \hat{V}(C_1, \sigma, (L, C_3, C_4)) &= 1.
 \end{aligned}$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

The verification is successful if the six verification formulas shown above all hold and fails if any one of the six formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_4 g_2, C_{2X})^{\frac{1}{x_i}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$m \cdot e(g_4 g_2, g)^r / e(g_4 g_2, C_{2X})^{\frac{1}{x_i}}$$

$$\begin{aligned} &= m \cdot e(g_4 g_2, g)^r / e(g_4 g_2, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4 g_2, g)^r / e(g_4 g_2, g)^r = m. \end{aligned}$$

(Fifth embodiment)

5 In the present embodiment, an example in which the server and users are honest entities and there is no possibility of collusion of a plurality of entities in the first embodiment will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and
10 decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

15 First, a public parameter generation unit 13 of a key generator 10 generates a public parameter based on a security parameter λ stored in a key parameter storage unit 11 in advance or acquires the parameter from outside (ST1).

$$\text{par} = \{\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, u, v, \text{Sig}\}$$

20 If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects an element x_i of Z_p^* from Z_p^* and sets a private key $sk_i = x_i$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $pk_i = X_i$ of the
25 re-encryption key generator 30 using the private key sk_i

and the public parameter par.

$$X_i = g^{x_i}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i = X_{1i}$ using the private key sk_i and the public parameter par (ST3).

$$X_{1i} = g_1^{x_i}$$

Similarly, if the identification information of a decryption device 50 is assumed to be j , the key generation unit 14 randomly selects an element x_j of Z_p^* from Z_p^* and sets a private key $sk_j = x_j$ of the decryption device 50. Then, the key generation unit 14 generates a public key $pk_j = X_j$ of the decryption device 50 using the private key sk_j and the public parameter par.

$$X_j = g^{x_j}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j = X_{1j}$ using the private key sk_j and the public parameter par (ST6).

$$X_{1j} = g_1^{x_j}$$

(2) Encryption processing

Like in step ST21 in the first embodiment, an encryption parameter generation unit 23 of an encryption device 20 generates (ssk, svk) (ST21) and sets the verification key svk to the encrypted data C_1 ($C_1 = svk$).

The encryption parameter generation unit 23 also

generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

5 The ciphertext generation unit 24 generates encrypted data C_2, C_3, C_4 below from a message $m \in G_T$ as plaintext data using the random number r , the public key pk_1 of the re-encryption key generator 30, and the public parameter par (ST22).

$$C_2 = X_i^r, C_3 = e(g_1, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

10 After step ST22 is completed, like in step ST23 in the first embodiment, the ciphertext generation unit 24 generates a one-time signature σ for the encrypted data C_3, C_4 based on a signature generation function $\hat{S}$ of the public parameter par and the signature key ssk generated in step ST21 (ST23).

15 Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_i = (C_1, C_2, C_3, C_4, \sigma)$ including all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext data into a temporary data storage unit 21.

20 A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

25 Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) is re-

encrypted into ciphertext addressed to the decryption
device 50 (in the example of FIG. 8, for example, the user
A) (that is, the transfer of decryption right of the
ciphertext to the user A). The decryption device 50
5 transmits the re-encryption key generation key rk_j of the
decryption device 50 to the re-encryption key generator 30.
A communication unit 33 of the re-encryption key generator
30 acquires and writes the re-encryption key generation key
 rk_j of the decryption device 50 into a temporary data
10 storage unit 32 under the control of a control unit 35
(ST31).

Because there is no need to generate a random number
for the generation of a re-encryption key in the present
embodiment, a random number generation unit 36 of the re-
15 encryption key generator 30 is not needed.

A re-encryption key generation unit 34 generates a re-
encryption key R_{ij} as shown in the following formulas using
the private key sk_i of the re-encryption key generator 30
in a private key storage unit 31, the re-encryption key
20 generation key rk_j of the decryption device 50 in the
temporary data storage unit 32, and the public parameter
par (ST32).

$$R_{ij} = X_{1j}^{1/x_i} = g_1^{\frac{x_j}{x_i}}$$

(4) Re-encryption processing

25 A re-encryption processing unit 44 of the re-
encryption device 40 verifies the ciphertext data C_1 in a

temporary data storage unit 42 using the public parameter
par and the public key pk_i of the re-encryption key
generator 30 based on the following verification formulas
(ST41).

$$e(C_2, u^{C_1} \cdot v) = e(X_i, C_4)$$

$$\hat{V}(C_1, \sigma, (C_3, C_4)) = 1.$$

5

The verification is successful if both verification
formulas shown above hold and fails if any one of the two
formulas does not hold.

If the verification is successful (or the verification
processing in step ST41 is skipped), a re-encryption
parameter generation unit 45 generates and sends out a
random number $t \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

The re-encryption processing unit 44 generates, as
shown in the following formulas, re-encrypted data C_2' ,
 C_2'' , C_2''' using the random number t , the ciphertext data
 C_i stored in the temporary data storage unit 42, the re-
encryption key R_{ij} , the public key pk_i of the re-encryption
key generator 30, and the public parameter par (ST42).

$$C_2' = X_i^t, C_2'' = R_{ij}^{1/t} = g_1^{\frac{x_j}{tx_i}}, C_2''' = C_2^t = X_i^{rt}$$

20

After step ST42 is completed, the re-encryption
processing unit 44 generates re-encrypted data
 $C_j = (C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ by replacing the encrypted
data C_2 in the ciphertext data C_i with all the above re-
encrypted data C_2' to C_2''' and writes the obtained re-

encrypted data C_j into the temporary data storage unit 42.

A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46 (ST43).

(5) Decryption processing

A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51 and the public parameter par in the temporary data storage unit 52 based on the following verification formulas (ST51).

$$\begin{aligned} e(C'_2, C''_2) &= e(g, g_1)^{x_j}, \\ e(C'''_2, u^{C_1} \cdot v) &= e(C'_2, C_4), \\ \hat{V}(C_1, \sigma, (C_3, C_4)) &= 1. \end{aligned} \quad \dots(c1)$$

Instead of the first verification formula in the formula (c1), the following verification formula may be used.

$$e(C'_2, C''_2) = e(X_j, g_1)$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

The verification is successful if the three verification formulas in the formula (c1) all hold and fails if any one of the three formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$m = C_3 / e(C_2'', C_2''')^{1/x_j}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} C_3 / e(C_2'', C_2''')^{1/x_j} &= C_3 / e(g_1^{\frac{x_j}{tx_i}}, g^{x_i rt})^{1/x_j} \\ &= m \cdot e(g_1, g)^r / e(g_1^{x_j}, g^r)^{1/x_j} = m. \end{aligned}$$

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ij} based on the re-encryption key generation key rk_j , the re-encryption key cannot be generated if the transfer destination (in the example of FIG. 8, for example, the user A) does not notify the transfer source (in the example of FIG. 8, the group administrator) of the re-encryption key generation key and therefore, re-encryption (that is, the transfer of ciphertext to the relevant transfer destination) without approval of the transfer destination can be prevented.

Also according to the present embodiment, as described above, due to the configuration of omitting g_2 to g_5 in the

first embodiment, the data size of the public parameter,
public key, re-encryption key generation key, re-encryption
key, private key, ciphertext data, and re-encrypted data
can be reduced when compared with the embodiment shown in
5 the first embodiment. Also according to the present
embodiment, the amount of computations needed for key setup
processing, encryption processing, re-encryption key
generation processing, re-encryption processing, and
decryption processing can be reduced when compared with the
10 embodiment shown in the first embodiment.

[Twelfth modification]

In the fifth embodiment, an example in which
ciphertext data is generated by the encryption device 20,
the ciphertext data is re-encrypted by the re-encryption
15 device 40 to generate re-encrypted data, and the re-
encrypted data is decrypted by the decryption device 50 is
cited, but the fifth embodiment may be changed to a form in
which the ciphertext data is decrypted without being re-
encrypted. In this case, only the key setup processing,
20 encryption processing, and decryption processing need to be
performed. The key setup processing in the present
modification is the same as the key setup processing in the
fifth embodiment. Hereinafter, the encryption processing
and the decryption processing in the present modification
25 will be described.

The difference between the encryption processing in
the present modification and the encryption processing in

the fifth embodiment is only in the last step. To provide a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formulas (ST51').

$$e\left(C_2, u^{C_1} \cdot v\right) = e(g, C_4)^{x_i}$$

$$\hat{V}(C_1, \sigma, (C_3, C_4)) = 1.$$

...(c2)

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

The verification is successful if both verification formulas in the formula (c2) hold and fails if any one of the two formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the

message m (ST52').

$$m = C_3 / e(C_2, g_1)^{1/x_i}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

5

$$\begin{aligned} & C_3 / e(g_1, C_2)^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^r / e(g_1, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^r / e(g_1, g)^r = m. \end{aligned}$$

[Thirteenth modification]

10 In addition to the twelfth modification, as will be described below, the fifth embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing and the decryption processing in the present modification

15 are the same as the key setup processing and the decryption processing in the fifth embodiment respectively.

Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

20 In the present modification, the identification information of the decryption device 50 is assumed to be j .

Like in step ST21 in the first embodiment, the encryption parameter generation unit 23 of the encryption

device 20 generates (ssk, svk) (ST21') and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

The encryption parameter generation unit 23 also generates and sends out two random numbers $r, t \in \mathbb{Z}_p^*$ to the
5 ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_2', C_2'', C_2''', C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random numbers r, t , the public key pk_j of the decryption device 50, and the public
10 parameter (ST22').

$$C_2' = X_j^t, C_2'' = g_1^{1/t}, C_2''' = X_j^{rt},$$

$$C_3 = e(g_1, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

After step ST22' is completed, the ciphertext generation unit 24 generates a one-time signature σ in the
15 same manner as in step ST23 (ST23').

Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_j = (C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ including all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext
20 data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit
25 (ST24').

25 The decryption device 50 verifies the ciphertext data

C_j generated by the encryption device 20 in the same manner as in step ST51 in the fifth embodiment (ST51').

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(C_2'', C_2''')^{1/x_j}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} C_3 / e(C_2'', C_2''')^{1/x_j} &= C_3 / e(g_1^{1/t}, g^{x_j r t})^{1/x_j} \\ &= m \cdot e(g_1, g)^r / e(g_1, g^r) = m. \end{aligned}$$

(Sixth embodiment)

In the present embodiment, an example of updating a re-encryption key in the fifth embodiment at certain intervals will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment or second embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

A public parameter generation unit 13 of a key

generator 10 generates a public parameter par in the same manner as in the fifth embodiment or acquires the parameter from outside (ST1). If the identification information of a re-encryption key generator 30 is i, a key generation unit 5 14 randomly selects elements x_i, y_i of Z_p^* from Z_p^* and sets a private key $sk_i=(x_i, y_i)$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $pk_i=(X_i, Y_i)$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par.

10
$$X_i = g^{x_i}, \quad Y_i = g^{y_i}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i=X_{1i}$ using the private key sk_i and the public parameter par (ST3).

$$X_{1i} = g_1^{x_i}$$

15 Similarly, if the identification information of a decryption device 50 is assumed to be j, the key generation unit 14 randomly selects elements x_j, y_j of Z_p^* from Z_p^* and sets a private key $sk_j=(x_j, y_j)$ of the decryption device 50. Then, the key generation unit 14 generates a public key 20 $pk_j=(X_j, Y_j)$ of the decryption device 50 using the private key sk_j and the public parameter par.

$$X_j = g^{x_j}, \quad Y_j = g^{y_j}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j=X_{1j}$ using the private key 25 sk_j and the public parameter par (ST6).

$$X_{1j} = g_1^{x_j}$$

(2) Encryption processing

Like in step ST21 in the first embodiment, an encryption parameter generation unit 23 of an encryption device 20 generates (ssk, svk) (ST21) and sets the verification key svk to the encrypted data C_0 ($C_0 = \text{svk}$).

The encryption parameter generation unit 23 also generates and sends out a random number $s \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data C_1, C_2, C_3, C_4 below from a message $m \in G_T$ as plaintext data using the random number s , the public key pk_i of the re-encryption key generator 30, the public parameter par , and a time parameter L (ST22).

$$C_1 = X_i^s, \quad C_2 = F_i(L)^s, \\ C_3 = e(g_1, g)^s \cdot m, \quad C_4 = (u^{\text{svk}} \cdot v)^s$$

The function $F_i(L)$ is defined as shown below.

$$F_i(L) = g^L \cdot Y_i = g^{L+y_i} \quad (L \in \mathbb{Z}_p^*)$$

After step ST22 is completed, the ciphertext generation unit 24 generates a one-time signature σ for the time parameter L and the encrypted data C_3, C_4 based on a signature generation function $\hat{S}$ of the public parameter par and the signature key ssk generated in step ST21 (ST23).

$$\sigma = \hat{S}(\text{ssk}, (L, C_3, C_4))$$

Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_i = (L, C_0, C_1, C_2, C_3, C_4, \sigma)$ including the time parameter L , all the above encrypted data C_0 to C_4 and the one-time signature σ and writes the obtained
5 ciphertext data into a temporary data storage unit 21.

A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

10 Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) and generated by using L as the time parameter value is re-
15 encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the
20 decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35
25 (ST31).

A random number generation unit 36 generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a re-encryption key generation

unit 34.

The re-encryption key generation unit 34 generates a re-encryption key R_{ijL} as shown in the following formulas using the random number r , the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, the public parameter par , and the time parameter L (ST32).

$$R_{ijL} = (A_{ijL}, B_{ijL})$$

$$A_{ijL} = X_{1j}^{1/x_i} \cdot (g^{L+y_i})^r, \quad B_{ijL} = g^{x_i r}$$

(4) Re-encryption processing

A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par , the time parameter L contained in the ciphertext data C_i in the temporary data storage unit 42, and the public key pk_i of the re-encryption key generator 30 based on the following verification formulas (ST41).

$$e(X_i, C_4) = e(C_1, u^{C_0} \cdot v),$$

$$e(X_i, C_2) = e(C_1, F_i(L)),$$

$$\hat{V}(C_0, \sigma, (L, C_3, C_4)) = 1.$$

The verification is successful if the three verification formulas shown above all hold and fails if any one of the three formulas does not hold.

If the verification is successful (or the verification

processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out two random numbers $t_1, t_2 \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

5 The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_1' , C_1'' , C_1''' , C_2' , C_2'' , C_2''' using the random numbers t_1, t_2 , the ciphertext data C_i stored in the temporary data storage unit 42, the re-encryption key R_{ijL} , the public key pk_i of
10 the re-encryption key generator 30, and the public parameter par (ST42).

$$C_1' = X_i^{t_1}, C_1'' = (A_{ijL})^{1/t_1}, C_1''' = C_1^{t_1} = X_i^{st_1},$$
$$C_2' = F_i(L)^{t_2}, C_2'' = B_{ijL}^{1/t_2}, C_2''' = C_2^{t_2} = F_i(L)^{st_2}$$

 After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data
15 $C_j = (L, C_0, C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_3, C_4, \sigma)$ by replacing the encrypted data C_1, C_2 in the ciphertext data C_i with all the above re-encrypted data C_1' to C_2''' and writes the obtained ciphertext data into the temporary data storage unit 42.

20 A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46 (ST43).

(5) Decryption processing

25 A decryption processing unit 54 of the decryption

device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51 and the public parameter par in the temporary data storage unit 52 based on the following verification formulas (ST51).

$$\begin{aligned} e(C_1''', u^{C_0} \cdot v) &= e(C_1', C_4), \\ e(C_2''', u^{C_0} \cdot v) &= e(C_2', C_4), \\ e(C_1', C_1'') &= e(g, g_1)^{x_j} \cdot e(C_2', C_2''), \\ \hat{V}(C_0, \sigma, (L, C_3, C_4)) &= 1. \end{aligned} \quad \dots(c3)$$

Instead of the third verification formula in the formula (c3), the following verification formula may be used.

$$e(C_1', C_1'') = e(X_j, g_1) \cdot e(C_2', C_2'')$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

The verification is successful if the four verification formulas in the formula (c3) all hold and fails if any one of the four formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$m = C_3 \cdot (e(C_2'', C_2''')/e(C_1'', C_1'''))^{1/x_j}.$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 \cdot \{e(g^{x_i r/t_2}, F_i(L)^{st_2})/e((g_1^{x_j/x_i} \cdot F_i(L)^r)^{1/t_1}, g^{sx_i t_1})\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_i r}, F_i(L)^s)/e(g_1^{x_j/x_i} \cdot F_i(L)^r, g^{sx_i})\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_i r}, F_i(L)^s)/e(g_1^{x_j} \cdot F_i(L)^{x_i r}, g^s)\}^{1/x_j} \\ &= m \cdot e(g_1, g)^s \cdot \{1/e(g_1^{x_j}, g^s)\}^{1/x_j} = m. \end{aligned}$$

5

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ijL} based on the random number r , the re-encryption key generation key rk_j , and the time parameter L , in addition to the effects of the fifth embodiment, whether to transfer the decryption right can be decided in each period so that finer access control can be exercised.

10

Accordingly, even if the decryption right of ciphertext for the transfer source (in the example of FIG. 8, the group administrator) is once transferred to the transfer

15

destination (in the example of FIG. 8, for example, the user A) in a certain period, it is possible not to grant the decryption right of the ciphertext for the transfer source to the relevant transfer destination, that is, the

20

decryption right of the transfer destination (of the ciphertext for the relevant transfer source) can be invalidated in the next period and therefore, a file sharing system whose convenience is further improved can be

provided.

Also according to the present embodiment, as described above, due to the configuration of omitting g_2 to g_5 in the second embodiment, the data size of the public parameter, public key, re-encryption key generation key, re-encryption key, private key, ciphertext data, and re-encrypted data can be reduced when compared with the embodiment shown in the second embodiment. Also according to the present embodiment, the amount of computations needed for key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing can be reduced when compared with the embodiment shown in the second embodiment.

[Fourteenth modification]

In the sixth embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is cited, but the sixth embodiment may be changed to a form in which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present modification is the same as the key setup processing in the sixth embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification

will be described.

The difference between the encryption processing in the present modification and the encryption processing in the sixth embodiment is only in the last step. To provide
5 a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data
10 storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device
15 50 based on the following verification formulas (ST51').

$$\begin{aligned} e(g, C_4)^{x_i} &= e(C_1, u^{C_0} \cdot v), \\ e(g, C_2)^{x_i} &= e(C_1, g)^{L+y_i}, \\ \hat{V}(C_0, \sigma, (L, C_3, C_4)) &= 1. \end{aligned} \quad \dots (c4)$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of
20 the local device 50 in the same manner as in step ST41.

The verification is successful if the three verification formulas in the formula (c4) all hold and fails if any one of the three formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(C_1, g_1)^{\frac{1}{x_i}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 / e(C_1, g_1)^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^s / e(g^{x_i s}, g_1)^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^s / e(g, g_1)^s = m. \end{aligned}$$

[Fifteenth modification]

In addition to the fourteenth modification, as will be described below, the sixth embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the sixth embodiment respectively.

Hereinafter, the encryption processing and the decryption

processing in the present modification will be described.
In the present modification, the identification information
of the decryption device 50 is assumed to be j .

5 Like in step ST21 in the first embodiment, the
encryption parameter generation unit 23 of the encryption
device 20 generates (ssk, svk) (ST21') and sets the
verification key svk to the encrypted data C_0 ($C_0=svk$).

10 The encryption parameter generation unit 23 also
generates and sends out two random numbers $t_1, t_2 \in \mathbb{Z}_p^*$ to the
ciphertext generation unit 24.

15 The ciphertext generation unit 24 generates encrypted
data $C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_3, C_4$ below from a
message $m \in G_T$ as plaintext data using the random numbers $t_1,$
 t_2 , the public key pk_j of the decryption device 50, the
public parameter, and the time parameter L (ST22').

$$\begin{aligned} C_1' &= X_j^{t_1}, C_1'' = (F_j(L) \cdot g_1)^{1/t_1}, C_1''' = X_j^{st_1}, \\ C_2' &= F_j(L)^{t_2}, C_2'' = X_j^{1/t_2}, C_2''' = F_j(L)^{st_2}, \\ C_3 &= e(g_1, g)^s \cdot m, C_4 = (u^{svk} \cdot v)^s \end{aligned}$$

After step ST22' is completed, the ciphertext
generation unit 24 generates a one-time signature σ in the
same manner as in step ST23.

20 Thereafter, the ciphertext generation unit 24
generates ciphertext data
 $C_j = (L, C_0, C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_3, C_4, \sigma)$ including the
time parameter L , all the above encrypted data C_0 to C_4 and
the one-time signature σ and writes the obtained ciphertext

data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_j generated by the encryption device 20 in the same manner as in step ST51 (ST51'). If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j and the public parameter par to obtain the message m (ST52').

$$m = C_3 \cdot (e(C_2'', C_2''')/e(C_1'', C_1'''))^{1/x_j}.$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 \cdot \{e(C_2'', C_2''')/e(C_1'', C_1''')\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_j/t_2}, F_j(L)^{st_2})/e((F_j(L) \cdot g_1)^{1/t_1}, g^{sx_j t_1})\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_j}, F_j(L)^s)/e(F_j(L) \cdot g_1, g^{sx_j})\}^{1/x_j} \\ &= m \cdot e(g_1, g)^s \cdot \{1/e(g_1, g^{sx_j})\}^{1/x_j} = m. \end{aligned}$$

(Seventh embodiment)

In the present embodiment, an example of omitting a one-time signature in the first embodiment will be

described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

First, a public parameter generation unit 13 of a key generator 10 generates a public parameter

$$\text{par} = \{\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_2, g_3, g_4, g_5, u, v, H\}$$

based on a security parameter λ stored in a key parameter storage unit 11 in advance or acquires the parameter from outside (ST1).

Here, g_3 is rewritten and denoted by g^n , g_4 is rewritten and denoted by g_1^n , and g_5 is rewritten and denoted by g_2^n . H is a hash function defined by the following formula.

$$H : \mathbb{G}^4 \times \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$$

If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects elements x_i, y_i, z_i of $\mathbb{Z}_p^*$ from $\mathbb{Z}_p^*$ and sets a private key $sk_i = (x_i, y_i, z_i)$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $pk_i = (X_i, Y_{1i}, Y_{2i}, Z_i, Z_{1i})$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par .

$$X_i = g^{X_i}, \quad Y_{1i} = g_1^{Y_i}, \quad Y_{2i} = g_2^{Y_i}, \\ Z_i = g^{Z_i}, \quad Z_{1i} = g_1^{Z_i}$$

If verification processing using a formula (d3) described later is not performed, data Y_{2i} may be omitted. That is, the public key pk_i only needs to include data X_i ,
5 Y_{1i} , Z_i , Z_{1i} .

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i = (X_{3i}, Y_{5i})$ using the private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{X_i}, \quad Y_{5i} = g_5^{Y_i}$$

10 Similarly, if the identification information of a decryption device 50 is assumed to be j , the key generation unit 14 randomly selects elements x_j , y_j , z_j of Z_p^* from Z_p^* and sets a private key $sk_j = (x_j, y_j, z_j)$ of the decryption device 50. Then, the key generation unit 14 generates a
15 public key $pk_j = (X_j, Y_{1j}, Y_{2j}, Z_j, Z_{1j})$ of the decryption device 50 using the private key sk_j and the public parameter par .

$$X_j = g^{X_j}, \quad Y_{1j} = g_1^{Y_j}, \quad Y_{2j} = g_2^{Y_j}, \\ Z_j = g^{Z_j}, \quad Z_{1j} = g_1^{Z_j}$$

If, similarly as described above, the verification processing using a formula (d3) described later is not
20 performed, data Y_{2j} may be omitted. That is, the public key pk_j only needs to include data X_j , Y_{1j} , Z_j , Z_{1j} .

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j=(X_{3j},Y_{5j})$ using the private key sk_j and the public parameter par (ST6).

$$X_{3j}=g_3^{X_j}, \quad Y_{5j}=g_5^{Y_j}$$

5 (2) Encryption processing

No one-time signature is used in the present embodiment and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed.

10 An encryption parameter generation unit 23 of an encryption device 20 generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

15 The ciphertext generation unit 24 generates encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random number r , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST22).

$$\begin{aligned} C_{2X} &= X_i^r, \quad C_{2Y} = Y_{1i}^r, \quad C_{2Z} = Z_i^r, \quad C_{2Z1} = Z_{1i}^r, \\ C_3 &= e(g_4 g_5, g)^r \cdot m, \\ C_4 &= \left(u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v \right)^r \end{aligned}$$

20 After step ST22 is completed, the ciphertext generation unit 24 generates ciphertext data $C_i=(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4)$ including all the above encrypted data C_{2X} to C_4 and writes the obtained ciphertext data into a temporary data storage unit 21.

25 A communication unit 22 transmits the ciphertext data

C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35 (ST31).

A random number generation unit 36 generates and sends out a random number $\theta \in \mathbb{Z}_p^*$ to a re-encryption key generation unit 34.

The re-encryption key generation unit 34 generates a re-encryption key R_{ij} as shown in the following formulas using the random number θ , the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32,

and the public parameter par (ST32).

$R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$

$$R_{ij1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} = g^{\frac{\pi(x_j + \theta)}{x_i}},$$

$$R_{ij2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} = g^{\frac{\pi(\beta y_j + \theta)}{y_i}},$$

$$R_{ij3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j + \beta y_j + \theta)}{z_i}}.$$

5

In the above formulas, g_2 is rewritten and denoted by g^β .

(4) Re-encryption processing

A re-encryption processing unit 44 of the re-
 10 encryption device 40 verifies the ciphertext data C_i in a
 temporary data storage unit 42 using the public parameter
 par and the public key pk_i of the re-encryption key
 generator 30 based on the following verification formulas
 (ST41).

$$e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) = e(X_i, C_4),$$

$$e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) = e(Y_{1i}, C_4),$$

$$e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) = e(Z_i, C_4),$$

$$e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) = e(Z_{1i}, C_4).$$

15

The verification is successful if the four
 verification formulas shown above all hold and fails if any
 one of the four formulas does not hold.

If the verification is successful (or the verification processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out three random numbers $s, t, k \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' , C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{5X} , C_{5Y} , C_{5Z} using these random numbers s, t, k , the ciphertext data C_i stored in the temporary data storage unit 42, the re-encryption key R_{ij} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$\begin{aligned} C_{2X}' &= X_i^s, C_{2X}'' = C_{2X}^s = X_i^{rs}, C_{2Y}' = Y_{1i}^t, C_{2Y}'' = C_{2Y}^t = Y_{1i}^{rt}, \\ C_{2Z}' &= Z_i^k, C_{2Z}'' = C_{2Z}^k = Z_i^{rk}, C_{2Z1}' = Z_{1i}^k, C_{2Z1}'' = C_{2Z1}^k = Z_{1i}^{rk}, \\ C_{5X} &= R_{ij1}^{\frac{1}{s}}, C_{5Y} = R_{ij2}^{\frac{1}{t}}, C_{5Z} = R_{ij3}^{\frac{1}{k}} \end{aligned}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data $C_j = (C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, C_{5X}, C_{5Y}, C_{5Z}, i)$ by adding all the above re-encrypted data C_{2X}' to C_{5Z} and the identification information i of the re-encryption key generator 30 to the ciphertext data C_i and writes the obtained re-encrypted data C_j into the temporary data storage unit 42.

A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46

(ST43).

(5) Decryption processing

A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51, the public parameter par in the temporary data storage unit 52, and the public key pk_i of the re-encryption key generator 30 identified by the identification information i of the re-encryption key generator 30 contained in the re-encrypted data C_j in the temporary data storage unit 52 based on the following verification formulas (ST51).

$$\begin{aligned} e(C''_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2X}, C_4), \\ e(C''_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2Y}, C_4), \\ e(C''_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2Z}, C_4), \\ e(C''_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2Z1}, C_4), \\ e(C_{5Z}, C'_{2Z}) &= e(C_{5X}, C'_{2X}) \cdot e(g_2, g_3)^{y_j}, \\ e(C_{5Z}, C'_{2Z1}) &= e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j}, \\ e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(X_i, C_4), \\ e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Y_{1i}, C_4), \\ e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_i, C_4), \\ e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_{1i}, C_4). \end{aligned} \dots (d1)$$

Instead of the fifth verification formula in the formula (d1), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) = e(C_{5X}, C'_{2X}) \cdot e(g_5, g)^{y_j}. \quad \dots (d2)$$

Also, instead of the fifth verification formula in the formula (d1), the following verification formula may be used.

5
$$e(C_{5Z}, C'_{2Z}) = e(C_{5X}, C'_{2X}) \cdot e(Y_{2j}, g_3). \quad \dots (d3)$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

Also, instead of the sixth verification formula in the
10 formula (d1), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(g_3, g_1)^{x_j}. \quad \dots (d4)$$

Also, instead of the sixth verification formula in the
15 formula (d1), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4). \quad \dots (d5)$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

20 The verification is successful if the 10 verification formulas in the formula (d1) all hold and fails if any one of the 10 formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption
25 processing unit 54 decrypts the re-encrypted data C_j as

shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

5

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{(\pi x_j + \beta y_j + \theta)/z_i k}, g^{z_i r k})}{e(g^{(\pi x_j + \theta)/x_i s}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{(\pi x_j + \beta y_j + \theta)/z_i k}, g^{\alpha z_i r k})}{e(g^{(\beta y_j + \theta)/y_i t}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^r)}{e(g^{\pi x_j + \theta}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^{\alpha r})}{e(g^{\beta y_j + \theta}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\ &= m \cdot e(g_4 g_2, g)^r / (e(g, g)^{\beta r} \cdot e(g, g)^{\pi \alpha r}) = m. \end{aligned}$$

15

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ij} based on the random number θ and the re-encryption key generation key rk_j , effects similar to those in the first embodiment can be obtained.

Also according to the present embodiment, as described

above, due to the configuration of omitting the one-time signature σ in the first embodiment, the data size of ciphertext data and re-encrypted data can be reduced when compared with the embodiment shown in the first embodiment.

5 Also according to the present embodiment, the amount of computations needed for encryption processing and re-encryption processing can be reduced when compared with the embodiment shown in the first embodiment.

[Sixteenth modification]

10 In the seventh embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is
15 cited, but the seventh embodiment may be changed to a form in which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present
20 modification is the same as the key setup processing in the seventh embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

25 The difference between the encryption processing in the present modification and the encryption processing in the seventh embodiment is only in the last step. To provide a description by making use of the above symbols,

the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formulas (ST51').

$$\begin{aligned} e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g, C_4)^{x_i}, \\ e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g_1, C_4)^{y_i}, \\ e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g, C_4)^{z_i}, \\ e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g_1, C_4)^{z_i}. \end{aligned} \quad \dots (d6)$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

The verification is successful if the four verification formulas in the formula (d6) all hold and fails if any one of the four formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local

device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_4g_5, C_{2X})^{\frac{1}{x_i}}$$

5 That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_4g_5, g)^r / e(g_4g_5, C_{2X})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_5, g)^r / e(g_4g_5, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_5, g)^r / e(g_4g_5, g)^r = m. \end{aligned}$$

[Seventeenth modification]

10 In addition to the sixteenth modification, as will be described below, the seventh embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption
15 processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the seventh embodiment respectively. Hereinafter, the encryption processing and the decryption
20 processing in the present modification will be described. In the present modification, the identification information of the decryption device 50 is assumed to be j.

No one-time signature is used in the present embodiment either and therefore, processing corresponding

to step ST21 and step ST23 in the first embodiment is not performed. The encryption parameter generation unit 23 of the encryption device 20 generates and sends out five random numbers $r, s, t, k, \theta \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, C_{5X}, C_{5Y}, C_{5Z}$ below from a message $m \in G_T$ as plaintext data using the random numbers r, s, t, k, θ , the public key pk_j of the decryption device 50, and the public parameter (ST22').

$$\begin{aligned} C_{2X}' &= Y_{2j}^s, C_{2X}'' = Y_{2j}^{rs}, C_{2Y}' = X_j^t, C_{2Y}'' = X_j^{rt}, \\ C_{2Z}' &= Y_{2j}^k, C_{2Z}'' = Y_{2j}^{rk}, C_{2Z1}' = X_j^k, C_{2Z1}'' = X_j^{rk}, \\ C_{2X} &= X_j^r, C_{2Y} = Y_{1j}^r, C_{2Z} = Z_j^r, C_{2Z1} = Z_{1j}^r, \\ C_3 &= e(g_4 g_5, g)^r \cdot m, C_4 = (u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v)^r, \\ C_{5X} &= (g_4 \cdot g_3^\theta)^{\frac{1}{s}}, C_{5Y} = g_3^{\frac{\theta+1}{t}}, C_{5Z} = (g_4 \cdot g_3^{\theta+1})^{\frac{1}{k}} \end{aligned}$$

After step ST22' is completed, the ciphertext generation unit 24 generates ciphertext data

$C_j = (C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, C_{5X}, C_{5Y}, C_{5Z}, j)$ including all the above encrypted data C_{2X}' to C_{5Z} and the identification information j of the decryption device 50 and writes the obtained ciphertext data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the

decryption device 50 under the control of the control unit
25 (ST24').

5 The decryption processing unit 54 of the decryption
device 50 verifies the re-encrypted data C_j in the
temporary data storage unit 52 using the private key sk_j of
the local device 50 in the private key storage unit 51 and
the public parameter par in the temporary data storage unit
52 based on the following verification formulas (formula
(d7)) (ST51').

10 In step ST51', as will be described later, the
decryption processing unit 54 may use the public key pk_j of
the decryption device 50 identified by the identification
information j of the re-encryption key degeneration device
30 contained in the re-encrypted data C_j in the temporary
15 data storage unit 52. This also applies to embodiments and
modifications described later.

$$\begin{aligned}
e(C_{2X}'', u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C_{2X}', C_4), \\
e(C_{2Y}'', u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C_{2Y}', C_4), \\
e(C_{2Z}'', u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C_{2Z}', C_4), \\
e(C_{2Z1}'', u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C_{2Z1}', C_4), \\
e(C_{5Z}, C_{2Z}') &= e(C_{5X}, C_{2X}') \cdot e(g_2, g_3)^{y_j}, \\
e(C_{5Z}, C_{2Z1}') &= e(C_{5Y}, C_{2Y}') \cdot e(g, g_4)^{x_j}, \\
e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g, C_4)^{x_j}, \\
e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g_1, C_4)^{y_j}, \\
e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g, C_4)^{z_j}, \\
e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g_1, C_4)^{z_j}. \dots (d7)
\end{aligned}$$

Instead of the fifth verification formula in the formula (d7), the verification formula shown in the above-mentioned formula (d2) or formula (d3) may be used. Also, instead of the sixth verification formula in the formula (d7), the verification formula shown in the above-mentioned formula (d4) or formula (d5) may be used. Also, instead of the seventh to tenth verification formulas in the formula (d7), the following verification formulas may be used.

$$\begin{aligned}
e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(X_j, C_4), \\
e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Y_{1j}, C_4), \\
e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_j, C_4), \\
e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_{1j}, C_4).
\end{aligned}$$

The verification is successful if the 10 verification formulas in the formula (d7) all hold and fails if any one of the 10 formulas does not hold.

5 If the verification is successful (or the verification processing in step ST51' is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 10 52 to obtain the message m (ST52').

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation. In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β . 15

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi(1+\alpha+\theta)/k}, g^{\beta y_j r k})}{e(g^{\pi(\alpha+\theta)/s}, g^{\beta y_j r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi(1+\alpha+\theta)/k}, g^{x_j r k})}{e(g^{\pi(\theta+1)/t}, g^{x_j r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi(1+\alpha+\theta)}, g^{\beta r})}{e(g^{\pi(\alpha+\theta)}, g^{\beta r})} \right) \cdot \left(\frac{e(g^{\pi(1+\alpha+\theta)}, g^r)}{e(g^{\pi(\theta+1)}, g^r)} \right) \right\} \end{aligned}$$

$$= m \cdot e(g_4 g_5, g)^r / (e(g, g)^{\pi \beta r} \cdot e(g, g)^{\pi \alpha r}) = m.$$

(Eighth embodiment)

In the present embodiment, an example of updating a re-encryption key in the seventh embodiment at certain intervals will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment or second embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

A public parameter generation unit 13 of a key generator 10 generates a public parameter par in the same manner as in the seventh embodiment or acquires the parameter from outside (ST1). H is a hash function defined by the following formula.

$$H : \mathbb{Z}_p^* \times \mathbb{G}^5 \times \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$$

If the identification information of a re-encryption key generator 30 is i, a key generation unit 14 randomly selects elements x_i, y_i, z_i, w_i of $\mathbb{Z}_p^*$ from $\mathbb{Z}_p^*$ and sets a private key $sk_i = (x_i, y_i, z_i, w_i)$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $pk_i = (X_i, Y_{1i}, Y_{2i}, Z_i, Z_{1i}, W_i)$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par.

$$\begin{aligned} X_i &= g^{x_i}, \quad Y_{1i} = g_1^{y_i}, \quad Y_{2i} = g_2^{y_i}, \\ Z_i &= g^{z_i}, \quad Z_{1i} = g_1^{z_i}, \quad W_i = g^{w_i} \end{aligned}$$

If verification processing using a formula (d10) described later is not performed, data Y_{2i} may be omitted. That is, the public key pk_i only needs to include data X_i , Y_{1i} , Z_i , Z_{1i} , W_i .

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i = (X_{3i}, Y_{5i})$ using the private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{x_i}, \quad Y_{5i} = g_5^{y_i}$$

Similarly, if the identification information of a decryption device 50 is assumed to be j , the key generation unit 14 randomly selects elements x_j, y_j, z_j, w_j of Z_p^* from Z_p^* and sets a private key $sk_j = (x_j, y_j, z_j, w_j)$ of the decryption device 50. Then, the key generation unit 14 generates a public key $pk_j = (X_j, Y_{1j}, Y_{2j}, Z_j, Z_{1j}, W_j)$ of the decryption device 50 using the private key sk_j and the public parameter par .

$$\begin{aligned} X_j &= g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Y_{2j} = g_2^{y_j}, \\ Z_j &= g^{z_j}, \quad Z_{1j} = g_1^{z_j}, \quad W_j = g^{w_j} \end{aligned}$$

If, similarly as described above, the verification processing using a formula (d10) described later is not performed, data Y_{2j} may be omitted. That is, the public key pk_j only needs to include data $X_j, Y_{1j}, Z_j, Z_{1j}, W_j$.

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j=(X_{3j},Y_{5j})$ using the private key sk_j and the public parameter par (ST6).

$$X_{3j}=g_3^{X_j}, \quad Y_{5j}=g_5^{Y_j}$$

5 (2) Encryption processing

No one-time signature is used in the present embodiment and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed.

10 An encryption parameter generation unit 23 of an encryption device 20 generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random number r , the public key pk_i of the re-encryption key generator 30, the public parameter par , and a time parameter L (ST22).

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, C_{2F} = F_i(L)^r, \\ C_3 = e(g_4g_5, g)^r \cdot m, C_4 = (u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v)^r$$

Incidentally, the ciphertext generation unit 24 may not include the time parameter L as input into the hash function H . That is, instead of $H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)$, $H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)$ may be calculated. This also applies to embodiments and modifications described later.

The function $F_i(L)$ is defined as shown below.

$$F_i(L) = g^L \cdot W_i = g^{L+w_i} \quad (L \in \mathbb{Z}_p^*)$$

After step ST22 is completed, the ciphertext generation unit 24 generates ciphertext data $C_i = (L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4)$ including the time parameter L and all the above encrypted data C_{2X} to C_4 and writes the obtained ciphertext data into a temporary data storage unit 21.

A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

10 (3) Re-encryption key generation processing

Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) and generated by using L as the time parameter value is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35 (ST31).

A random number generation unit 36 generates and sends

out three random numbers $\theta, \delta_x, \delta_y \in \mathbb{Z}_p^*$ to a re-encryption key generation unit 34.

The re-encryption key generation unit 34 generates a re-encryption key R_{ijL} as shown in the following formulas using the random numbers $\theta, \delta_x, \delta_y$, the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, the public parameter par , and the time parameter L (ST32).

10 $R_{ijL} = (R_{ijL1}, R_{ijL2}, R_{ijL3}, R_{ijL4}, R_{ijL5})$

$$R_{ijL1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} \cdot g^{(L+w_i)\delta_x} = g^{\frac{\pi(x_j+\theta)}{x_i} + (L+w_i)\delta_x},$$

$$R_{ijL2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} \cdot g^{(L+w_i)\delta_y} = g^{\frac{\pi(\beta y_j+\theta)}{y_i} + (L+w_i)\delta_y},$$

$$R_{ijL3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j+\beta y_j+\theta)}{z_i}},$$

$$R_{ijL4} = g^{x_i\delta_x}, R_{ijL5} = g_1^{y_i\delta_y} = g^{\alpha y_i\delta_y}$$

In the above formulas, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

(4) Re-encryption processing

15 A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par , the time parameter L contained in the ciphertext data C_i in the temporary data storage unit 42, and the public
20 key pk_i of the re-encryption key generator 30 based on the

following verification formulas (ST41).

$$\begin{aligned}e(C_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(X_i, C_4), \\e(C_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Y_{1i}, C_4), \\e(C_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_i, C_4), \\e(C_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_{1i}, C_4), \\e(C_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(F_i(L), C_4).\end{aligned}$$

If the time parameter L is not contained as input into the hash function H, instead of $H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)$,
5 $H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)$ is calculated for the verification. This also applies to verification processing of ciphertext data or re-encrypted data described below.

The verification is successful if the five
verification formulas shown above all hold and fails if any
10 one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out four random numbers $s, t, k, h \in \mathbb{Z}_p^*$ to the re-encryption processing
15 unit 44.

The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data $C_{2X'}$, $C_{2X''}$, $C_{2Y'}$, $C_{2Y''}$, $C_{2Z'}$, $C_{2Z''}$, $C_{2Z1'}$, $C_{2Z1''}$, $C_{2F'}$, $C_{2F''}$, C_{5X} , C_{5Y} , C_{5Z} , C_{5FX} , C_{5FY} using these random numbers s, t, k, h , the
20 ciphertext data C_i stored in the temporary data storage

unit 42, the re-encryption key R_{ijL} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$\begin{aligned} C'_{2X} &= X_i^s, C''_{2X} = C_{2X}^s = X_i^{rs}, C'_{2Y} = Y_{1i}^t, C''_{2Y} = C_{2Y}^t = Y_{1i}^{rt}, \\ C'_{2Z} &= Z_i^k, C''_{2Z} = C_{2Z}^k = Z_i^{rk}, C'_{2Z1} = Z_{1i}^k, C''_{2Z1} = C_{2Z1}^k = Z_{1i}^{rk}, \\ C'_{2F} &= F_i(L)^h, C''_{2F} = C_{2F}^h = F_i(L)^{rh}, \\ C_{5X} &= R_{ijL1}^{\frac{1}{s}}, C_{5Y} = R_{ijL2}^{\frac{1}{t}}, C_{5Z} = R_{ijL3}^{\frac{1}{k}}, \\ C_{5FX} &= R_{ijL4}^{\frac{1}{h}}, C_{5FY} = R_{ijL5}^{\frac{1}{h}} \end{aligned}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data

$$C_j = (L, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'',$$

$C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4, C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, i)$ by adding all the above re-encrypted data C_{2X}' to C_{5FY} and the identification information i of the re-encryption key generator 30 to the ciphertext data C_1 and writes the obtained ciphertext data into the temporary data storage unit 42.

A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46 (ST43).

(5) Decryption processing

A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local

device 50 in a private key storage unit 51, the public parameter par in the temporary data storage unit 52, and the public key pk_i of the re-encryption key generator 30 identified by the identification information i of the re-encryption key generator 30 contained in the re-encrypted data C_j in the temporary data storage unit 52 based on the following verification formulas (ST51).

$$\begin{aligned}
 e(C''_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2X}, C_4), \\
 e(C''_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2Y}, C_4), \\
 e(C''_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2Z}, C_4), \\
 e(C''_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2Z1}, C_4), \\
 e(C''_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2F}, C_4), \\
 e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) &= e(C_{5X}, C'_{2X}) \cdot e(g_2, g_3)^{y_j}, \\
 e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) &= e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j}, \\
 e(C_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(X_i, C_4), \\
 e(C_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Y_{1i}, C_4), \\
 e(C_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_i, C_4), \\
 e(C_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_{1i}, C_4), \\
 e(C_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(F_i(L), C_4). \dots (d8)
 \end{aligned}$$

Instead of the sixth verification formula in the formula (d8), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) = e(C_{5X}, C'_{2X}) \cdot e(g_5, g)^{y_j}. \dots (d9)$$

Also, instead of the sixth verification formula in the formula (d8), the following verification formula may be used.

$$e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) = e(C_{5X}, C'_{2X}) \cdot e(Y_{2j}, g_3). \dots (d10)$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

5 Instead of the seventh verification formula in the formula (d8), the verification formula (d11) or (d12) may be used.

$$e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) = e(C_{5Y}, C'_{2Y}) \cdot e(g_3, g_1)^{x_j} \dots (d11)$$

$$e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4) \dots (d12)$$

10 In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

The verification is successful if the 12 verification formulas in the formula (d8) all hold and fails if any one of the 12 formulas does not hold.

15 If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and
20 the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z}) \cdot e(C_{5FX}, C''_{2F})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1}) \cdot e(C_{5FY}, C''_{2F})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

25 In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned}
& C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z}) \cdot e(C_{5FX}, C''_{2F})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1}) \cdot e(C_{5FY}, C''_{2F})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\
& = C_3 / \left\{ \left(\frac{e(g^{\frac{\pi(x_j+\beta y_j+\theta)}{z_i k}}, g^{z_i r k}) \cdot e(g^{\frac{x_i \delta_x}{h}}, g^{(L+w_i) r h})}{e(g^{\frac{(\pi(x_j+\theta)/x_i)+(L+w_i)\delta_x}{s}}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\pi(x_j+\beta y_j+\theta)}{z_i k}}, g^{\alpha z_i r k}) \cdot e(g^{\frac{\alpha y_i \delta_y}{h}}, g^{(L+w_i) r h})}{e(g^{\frac{(\pi(\beta y_j+\theta)/y_i)+(L+w_i)\delta_y}{t}}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \\
& = C_3 / \left\{ \left(\frac{e(g^{\pi(x_j+\beta y_j+\theta)}, g^r) \cdot e(g^{x_i \delta_x}, g^{(L+w_i) r})}{e(g^{\pi(x_j+\theta)+(L+w_i)x_i \delta_x}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi(x_j+\beta y_j+\theta)}, g^{\alpha r}) \cdot e(g^{\alpha y_i \delta_y}, g^{(L+w_i) r})}{e(g^{\pi(\beta y_j+\theta)+(L+w_i)y_i \delta_y}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\
& = m \cdot e(g_4 g_5, g)^r / (e(g, g)^{\pi \beta r} \cdot e(g, g)^{\pi \alpha r}) = m.
\end{aligned}$$

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ijL} based on the random numbers θ , δ_x , δ_y , the re-encryption key generation key rk_j , and the time parameter L , in addition to the effects of the seventh embodiment, whether to transfer the decryption right can be decided in each period so that finer access control can be exercised. Accordingly, even if the decryption right of ciphertext for the transfer source (in the example of FIG. 8, the group administrator) is once transferred to the transfer destination (in the example of FIG. 8, for example, the user A) in a certain period, it is possible not to grant the decryption right of the ciphertext for the transfer source to the relevant transfer destination, that is, the decryption right of the transfer destination (of the ciphertext for the relevant transfer source) can be invalidated in the next period and therefore, a file

sharing system whose convenience is further improved can be provided.

Also according to the present embodiment, as described above, due to the configuration of omitting the one-time signature σ in the second embodiment, the data size of ciphertext data and re-encrypted data can be reduced when compared with the embodiment shown in the second embodiment. Also according to the present embodiment, the amount of computations needed for encryption processing and re-encryption processing can be reduced when compared with the embodiment shown in the second embodiment.

[Eighteenth modification]

In the eighth embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is cited, but the eighth embodiment may be changed to a form in which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present modification is the same as the key setup processing in the eighth embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

The difference between the encryption processing in

the present modification and the encryption processing in the eighth embodiment is only in the last step. To provide a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

10 The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formulas (ST51').

$$\begin{aligned} e(C_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{x_i}, \\ e(C_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g_1, C_4)^{y_i}, \\ e(C_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{z_i}, \\ e(C_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g_1, C_4)^{z_i}, \\ e(C_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{L+w_i}. \end{aligned}$$

... (d13)

15 Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

20 The verification is successful if the five verification formulas in the formula (d13) all hold and fails if any one of the five formulas does not hold.

 If the verification is successful (or the verification

processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_4g_5, C_{2X})^{\frac{1}{x_i}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_4g_5, g)^r / e(g_4g_5, C_{2X})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_5, g)^r / e(g_4g_5, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_5, g)^r / e(g_4g_5, g)^r = m. \end{aligned}$$

[Nineteenth modification]

In addition to the eighteenth modification, as will be described below, the eighth embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the eighth embodiment respectively.

Hereinafter, the encryption processing and the decryption processing in the present modification will be described. In the present modification, the identification information

of the decryption device 50 is assumed to be j .

No one-time signature is used in the present embodiment either and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed. The encryption parameter generation unit 23 of the encryption device 20 generates and sends out eight random numbers $r, s, t, k, h, \theta, \delta_x, \delta_y \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4, C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}$ below from a message $m \in G_T$ as plaintext data using the random numbers $r, s, t, k, h, \theta, \delta_x, \delta_y$, the public key pk_j of the decryption device 50, the public parameter, and the time parameter L (ST22').

$$\begin{aligned} C_{2X}' &= Y_{2j}^s, C_{2X}'' = Y_{2j}^{rs}, C_{2Y}' = X_j^t, C_{2Y}'' = X_j^{rt}, \\ C_{2Z}' &= Y_{2j}^k, C_{2Z}'' = Y_{2j}^{rk}, C_{2Z1}' = X_j^k, C_{2Z1}'' = X_j^{rk}, \\ C_{2F}' &= F_j(L)^h, C_{2F}'' = F_j(L)^{rh}, \\ C_{2X} &= X_j^r, C_{2Y} = Y_{1j}^r, C_{2Z} = Z_j^r, C_{2Z1} = Z_{1j}^r, C_{2F} = F_j(L)^r, \\ C_3 &= e(g_4 g_5, g)^r \cdot m, C_4 = (u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v)^r, \end{aligned}$$

$$C_{5X} = (g_4 \cdot g_3^\theta \cdot F_j(L)^{\delta_y})^{\frac{1}{s}} = g^{\frac{\pi(\alpha+\theta)+(L+w_j)\delta_y}{s}},$$

$$C_{5Y} = (g_3^{1+\theta} \cdot F_j(L)^{\delta_x})^{\frac{1}{t}} = g^{\frac{\pi(1+\theta)+(L+w_j)\delta_x}{t}},$$

$$C_{5Z} = (g_4 \cdot g_3^{1+\theta})^{\frac{1}{k}} = g^{\frac{\pi(\alpha+1+\theta)}{k}},$$

$$C_{5FX} = (Y_{2j})^{\frac{\delta_y}{h}}, C_{5FY} = (X_j)^{\frac{\delta_x}{h}}$$

After step ST22' is completed, the ciphertext generation unit 24 generates ciphertext data $C_j = (L, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4, C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, j)$ including the time parameter L, all the above encrypted data C_{2X}' to C_{5FY} and the identification information j of the decryption device 50 and writes the obtained ciphertext data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in the temporary data storage unit 52 using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 based on the following verification formulas (ST51').

$$\begin{aligned}
e(C''_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2X}, C_4), \\
e(C''_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2Y}, C_4), \\
e(C''_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2Z}, C_4), \\
e(C''_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2Z1}, C_4), \\
e(C''_{2F}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(C'_{2F}, C_4), \\
e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) &= e(C_{5X}, C'_{2X}) \cdot e(g_2, g_3)^{y_j}, \\
e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) &= e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j}, \\
e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{x_j}, \\
e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g_1, C_4)^{y_j}, \\
e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{z_j}, \\
e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g_1, C_4)^{z_j}, \\
e(C_{2F}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{L+w_j}.
\end{aligned}
\tag{d14}$$

5 Instead of the sixth verification formula in the formula (d14), the verification formula shown in the formula (d9) or (d10) may be used. Also, instead of the seventh verification formula in the formula (d14), the verification formula shown in the formula (d11) or (d12) may be used. Also, instead of the eighth to twelfth verification formulas in the formula (d14), the following verification formulas shown below may be used.

$$\begin{aligned}
 e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(X_j, C_4), \\
 e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Y_{1j}, C_4), \\
 e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_j, C_4), \\
 e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_{1j}, C_4), \\
 e(C_{2F}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(F_j(L), C_4).
 \end{aligned}$$

The verification is successful if the 12 verification formulas in the formula (d14) all hold and fails if any one of the 12 formulas does not hold.

5 If the verification is successful (or the verification processing in step ST51' is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and
10 the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52').

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

15 In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

$$\begin{aligned}
&= C_3 / \left\{ \left(\frac{e(g^{\frac{\pi(\alpha+1+\theta)}{k}}, g^{\beta y_j r k}) \cdot e(g^{\frac{\beta y_j \delta_y}{h}}, g^{(L+w_j) r h})}{e(g^{\frac{\pi(\alpha+\theta)+(L+w_j)\delta_y}{s}}, g^{\beta y_j r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\pi(\alpha+1+\theta)}{k}}, g^{x_j r k}) \cdot e(g^{\frac{x_j \delta_x}{h}}, g^{(L+w_j) r h})}{e(g^{\frac{\pi(1+\theta)+(L+w_j)\delta_x}{t}}, g^{x_j r t})} \right)^{\frac{1}{x_j}} \right\} \\
&= C_3 / \left\{ \left(\frac{e(g^{\pi(\alpha+1+\theta)}, g^{\beta r}) \cdot e(g^{\beta \delta_y}, g^{(L+w_j) r})}{e(g^{\pi(\alpha+\theta)+(L+w_j)\delta_y}, g^{\beta r})} \right) \cdot \left(\frac{e(g^{\pi(\alpha+1+\theta)}, g^r) \cdot e(g^{\delta_x}, g^{(L+w_j) r})}{e(g^{\pi(1+\theta)+(L+w_j)\delta_x}, g^r)} \right) \right\} \\
&= m \cdot e(g_4 g_5, g)^r / (e(g, g)^{\pi \beta r} \cdot e(g, g)^{\pi \alpha r}) = m.
\end{aligned}$$

(Ninth embodiment)

5 In the present embodiment, an example of omitting a one-time signature in the third embodiment will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the
10 present embodiment will be described. Portions common to the aforementioned first embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

15 First, a public parameter generation unit 13 of a key generator 10 generates a public parameter

$$\text{par} = \{\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, g_2, g_3, g_4, u, v, H\}$$

based on a security parameter λ stored in a key parameter storage unit 11 in advance or acquires the parameter from outside (ST1). Here, $g_3 = g^n$ and $g_4 = g_1^n$ are set. H is a hash
20 function defined by the following formula.

$$H : \mathbb{G}^4 \times \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$$

If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects elements x_i, y_i, z_i of Z_p^* from Z_p^* and sets a private key $sk_i=(x_i, y_i, z_i)$ of the re-encryption key

5 generator 30. Then, the key generation unit 14 generates a public key $pk_i=(X_i, Y_{1i}, Z_i, Z_{1i})$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par .

$$X_i = g^{x_i}, \quad Y_{1i} = g_1^{y_i}, \quad Z_i = g^{z_i}, \quad Z_{1i} = g_1^{z_i}$$

10 Further, the key generation unit 14 generates a re-encryption key generation key $rk_i=(X_{3i}, Y_{2i})$ using the private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{x_i}, \quad Y_{2i} = g_2^{y_i}$$

15 Similarly, if the identification information of a decryption device 50 is assumed to be j , the key generation unit 14 randomly selects elements x_j, y_j, z_j of Z_p^* from Z_p^* and sets a private key $sk_j=(x_j, y_j, z_j)$ of the decryption device 50. Then, the key generation unit 14 generates a public key $pk_j=(X_j, Y_{1j}, Z_j, Z_{1j})$ of the decryption device 50

20 using the private key sk_j and the public parameter par .

$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Z_j = g^{z_j}, \quad Z_{1j} = g_1^{z_j},$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j=(X_{3j}, Y_{2j})$ using the private key sk_j and the public parameter par (ST6).

$$X_{3j} = g_3^{x_j}, \quad Y_{2j} = g_2^{y_j}$$

25

(2) Encryption processing

No one-time signature is used in the present embodiment and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed.

5 An encryption parameter generation unit 23 of an encryption device 20 generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data C_{2X} , C_{2Y} , C_{2Z} , C_{2Z1} , C_3 , C_4 below from a message $m \in G_T$ as
10 plaintext data using the random number r , the public key pk_1 of the re-encryption key generator 30, and the public parameter par (ST22).

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_3 = e(g_4 g_2, g)^r \cdot m, C_4 = (u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v)^r$$

After step ST22 is completed, the ciphertext
15 generation unit 24 generates ciphertext data $C_i = (C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4)$ including all the above encrypted data C_{2X} to C_4 and writes the obtained ciphertext data into a temporary data storage unit 21.

A communication unit 22 transmits the ciphertext data
20 C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that
25 ciphertext addressed to the re-encryption key generator 30

(in the example of FIG. 8, the group administrator) is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35 (ST31).

A random number generation unit 36 generates and sends out a random number $\theta \in \mathbb{Z}_p^*$ to a re-encryption key generation unit 34.

The re-encryption key generation unit 34 generates a re-encryption key R_{ij} as shown in the following formulas using the random number θ , the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, and the public parameter par (ST32).

$$R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$$

$$R_{ij1} = (X_{3j} \cdot g^\theta)^{1/x_i} = g^{\frac{\pi x_j + \theta}{x_i}},$$

$$R_{ij2} = (Y_{2j} \cdot g^\theta)^{1/y_i} = g^{\frac{\beta y_j + \theta}{y_i}},$$

$$R_{ij3} = (X_{3j} \cdot Y_{2j} \cdot g^{\theta})^{1/z_i} = g^{\frac{\pi x_j + \beta y_j + \theta}{z_i}}.$$

In the above formulas, g_2 is rewritten and denoted by g^{β} .

(4) Re-encryption processing

5 A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par and the public key pk_i of the re-encryption key generator 30 based on the following verification formulas
10 (ST41).

$$\begin{aligned} e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(X_i, C_4), \\ e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Y_{1i}, C_4), \\ e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_i, C_4), \\ e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_{1i}, C_4). \end{aligned}$$

The verification is successful if the four verification formulas shown above all hold and fails if any one of the four formulas does not hold.

15 If the verification is successful (or the verification processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out three random numbers $s, t, k \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

20 The re-encryption processing unit 44 generates, as

shown in the following formulas, re-encrypted data C_{2X}' ,
 C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{5X} , C_{5Y} , C_{5Z}
using these random numbers s, t, k , the ciphertext data C_i
stored in the temporary data storage unit 42, the re-
5 encryption key R_{ij} , the public key pk_i of the re-encryption
key generator 30, and the public parameter par (ST42).

$$\begin{aligned} C_{2X}' &= X_i^s, C_{2X}'' = C_{2X}^s = X_i^{rs}, C_{2Y}' = Y_{1i}^t, C_{2Y}'' = C_{2Y}^t = Y_{1i}^{rt}, \\ C_{2Z}' &= Z_i^k, C_{2Z}'' = C_{2Z}^k = Z_i^{rk}, C_{2Z1}' = Z_{1i}^k, C_{2Z1}'' = C_{2Z1}^k = Z_{1i}^{rk}, \\ C_{5X} &= R_{ij1}^{\frac{1}{s}}, C_{5Y} = R_{ij2}^{\frac{1}{t}}, C_{5Z} = R_{ij3}^{\frac{1}{k}} \end{aligned}$$

After step ST42 is completed, the re-encryption
10 processing unit 44 generates re-encrypted data

$C_j = (C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{3}, C_{4}, C_{5X}, C_{5Y}, C_{5Z}, i)$ by adding all the above re-encrypted data
 C_{2X}' to C_{5Z} and the identification information i of the re-
encryption key generator 30 to the ciphertext data C_i and
15 writes the obtained re-encrypted data C_j into the temporary
data storage unit 42.

A communication unit 43 transmits the re-encrypted
data C_j in the temporary data storage unit 42 to the
decryption device 50 under the control of a control unit 46
20 (ST43).

(5) Decryption processing

A decryption processing unit 54 of the decryption
device 50 verifies the re-encrypted data C_j in a temporary
data storage unit 52 using the private key sk_j of the local
25 device 50 in a private key storage unit 51, the public

parameter par in the temporary data storage unit 52, and the public key pk_i of the re-encryption key generator 30 identified by the identification information i of the re-encryption key generator 30 contained in the re-encrypted data C_j in the temporary data storage unit 52 based on the following verification formulas (ST51).

5

$$\begin{aligned} e(C''_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2X}, C_4), \\ e(C''_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2Y}, C_4), \\ e(C''_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2Z}, C_4), \\ e(C''_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(C'_{2Z1}, C_4), \\ e(C_{5Z}, C'_{2Z}) &= e(C_{5X}, C'_{2X}) \cdot e(g_2, g)^{y_j}, \\ e(C_{5Z}, C'_{2Z1}) &= e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j}, \\ e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(X_i, C_4), \\ e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Y_{1i}, C_4), \\ e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_i, C_4), \\ e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(Z_{1i}, C_4). \end{aligned} \dots (e1)$$

Instead of the sixth verification formula in the formula (e1), the following verification formula may be used.

10

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4).$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

15

The verification is successful if the 10 verification

formulas in the formula (e1) all hold and fails if any one of the 10 formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{(\pi x_j + \beta y_j + \theta)/z_i k}, g^{z_i r k})}{e(g^{(\pi x_j + \theta)/x_i s}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{(\pi x_j + \beta y_j + \theta)/z_i k}, g^{\alpha z_i r k})}{e(g^{(\beta y_j + \theta)/y_i t}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^r)}{e(g^{\pi x_j + \theta}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^{\alpha r})}{e(g^{\beta y_j + \theta}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\ &= m \cdot e(g_4 g_2, g)^r / (e(g, g)^{\beta r} \cdot e(g, g)^{\pi \alpha r}) = m. \end{aligned}$$

According to the present embodiment, as described

above, due to the configuration of generating the re-encryption key R_{ij} based on the random number θ and the re-encryption key generation key rk_j , effects similar to those in the third embodiment can be obtained.

5 Also according to the present embodiment, as described above, due to the configuration of omitting the one-time signature σ in the third embodiment, the data size of ciphertext data and re-encrypted data can be reduced when compared with the embodiment shown in the third embodiment.

10 Also according to the present embodiment, the amount of computations needed for encryption processing and re-encryption processing can be reduced when compared with the embodiment shown in the third embodiment.

[Twentieth modification]

15 In the ninth embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is

20 cited, but the ninth embodiment may be changed to a form in which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present

25 modification is the same as the key setup processing in the ninth embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification

will be described.

The difference between the encryption processing in the present modification and the encryption processing in the ninth embodiment is only in the last step. To provide
5 a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data
10 storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device
15 50 based on the following verification formulas (ST51').

$$\begin{aligned} e(C_{2X}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g, C_4)^{x_i}, \\ e(C_{2Y}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g_1, C_4)^{y_i}, \\ e(C_{2Z}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g, C_4)^{z_i}, \\ e(C_{2Z1}, u^{H(C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)} \cdot v) &= e(g_1, C_4)^{z_i}. \end{aligned} \dots (e2)$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of
20 the local device 50 in the same manner as in step ST41.

The verification is successful if the four verification formulas in the formula (e2) all hold and fails if any one of the four formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_4g_2, C_{2X})^{\frac{1}{x_i}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_4g_2, g)^r / e(g_4g_2, C_{2X})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_2, g)^r / e(g_4g_2, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_4g_2, g)^r / e(g_4g_2, g)^r = m. \end{aligned}$$

(Tenth embodiment)

In the present embodiment, an example of updating a re-encryption key in the ninth embodiment at certain intervals will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment or second embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

A public parameter generation unit 13 of a key generator 10 generates a public parameter par in the same

manner as in the ninth embodiment or acquires the parameter from outside (ST1). H is a hash function defined by the following formula.

$$H : \mathbb{Z}_p^* \times \mathbb{G}^5 \times \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$$

5 If the identification information of a re-encryption key generator 30 is i, a key generation unit 14 randomly selects elements x_i, y_i, z_i, w_i of $\mathbb{Z}_p^*$ from $\mathbb{Z}_p^*$ and sets a private key $sk_i=(x_i, y_i, z_i, w_i)$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a
10 public key $pk_i=(X_i, Y_{1i}, Z_i, Z_{1i}, W_i)$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par.

$$X_i = g^{x_i}, \quad Y_{1i} = g_1^{y_i}, \quad Z_i = g^{z_i}, \\ Z_{1i} = g_1^{z_i}, \quad W_i = g^{w_i}$$

15 Further, the key generation unit 14 generates a re-encryption key generation key $rk_i=(X_{3i}, Y_{2i})$ using the private key sk_i and the public parameter par (ST3).

$$X_{3i} = g_3^{x_i}, \quad Y_{2i} = g_2^{y_i}$$

20 Similarly, if the identification information of a decryption device 50 is assumed to be j, the key generation unit 14 randomly selects elements x_j, y_j, z_j, w_j of $\mathbb{Z}_p^*$ from $\mathbb{Z}_p^*$ and sets a private key $sk_j=(x_j, y_j, z_j, w_j)$ of the decryption device 50. Then, the key generation unit 14 generates a public key $pk_j=(X_j, Y_{1j}, Z_j, Z_{1j}, W_j)$ of the decryption device 50 using the private key sk_j and the
25 public parameter par.

$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Z_j = g^{z_j},$$

$$Z_{1j} = g_1^{z_j}, \quad W_j = g^{w_j}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j = (X_{3j}, Y_{2j})$ using the private key sk_j and the public parameter par (ST6).

5
$$X_{3j} = g_3^{x_j}, \quad Y_{2j} = g_2^{y_j}$$

(2) Encryption processing

No one-time signature is used in the present embodiment and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed.

10 An encryption parameter generation unit 23 of an encryption device 20 generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random number r , the public key pk_i of the re-encryption key generator 30, the public parameter par , and a time parameter L (ST22).

15
$$C_{2X} = X_i^r, \quad C_{2Y} = Y_{1i}^r, \quad C_{2Z} = Z_i^r, \quad C_{2Z1} = Z_{1i}^r,$$

$$C_{2F} = F_i(L)^r, \quad C_3 = e(g_4 g_2, g)^r \cdot m,$$

$$C_4 = (u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v)^r$$

20

The function $F_1(L)$ is defined as shown below.

$$F_i(L) = g^L \cdot W_i = g^{L+w_i} \quad (L \in \mathbb{Z}_p^*)$$

After step ST22 is completed, the ciphertext generation unit 24 generates ciphertext data

$C_i = (L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4)$ including the time parameter
5 L and all the above encrypted data C_{2X} to C_4 and writes the obtained ciphertext data into a temporary data storage unit 21.

A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to a re-encryption
10 device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30
15 (in the example of FIG. 8, the group administrator) and generated by using L as the time parameter value is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user A) (that is, the transfer of decryption right of the
20 ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator 30 acquires and writes the re-encryption key generation key
25 rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35 (ST31).

A random number generation unit 36 generates and sends out three random numbers $\theta, \delta_x, \delta_y \in \mathbb{Z}_p^*$ to a re-encryption key generation unit 34.

5 The re-encryption key generation unit 34 generates a re-encryption key R_{ijL} as shown in the following formulas using the random numbers $\theta, \delta_x, \delta_y$, the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, 10 the public parameter par , and the time parameter L (ST32).

$$R_{ijL} = (R_{ijL1}, R_{ijL2}, R_{ijL3}, R_{ijL4}, R_{ijL5})$$

$$R_{ijL1} = (X_{3j} \cdot g^\theta)^{1/x_i} \cdot g^{(L+w_i)\delta_x} = g^{\frac{\pi x_j + \theta}{x_i} + (L+w_i)\delta_x},$$

$$R_{ijL2} = (Y_{2j} \cdot g^\theta)^{1/y_i} \cdot g^{(L+w_i)\delta_y} = g^{\frac{\beta y_j + \theta}{y_i} + (L+w_i)\delta_y},$$

$$R_{ijL3} = (X_{3j} \cdot Y_{2j} \cdot g^\theta)^{1/z_i} = g^{\frac{\pi x_j + \beta y_j + \theta}{z_i}},$$

$$R_{ijL4} = g^{x_i \delta_x}, R_{ijL5} = g_1^{y_i \delta_y} = g^{\alpha y_i \delta_y}$$

In the above formulas, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

15 (4) Re-encryption processing

A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par , the time parameter L contained in the ciphertext data C_i in the temporary data storage unit 42, and the public key pk_i of the re-encryption key generator 30 based on the following verification formulas (ST41). 20

$$\begin{aligned}
 e(C_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(X_i, C_4), \\
 e(C_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Y_{1i}, C_4), \\
 e(C_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_i, C_4), \\
 e(C_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(Z_{1i}, C_4), \\
 e(C_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(F_i(L), C_4).
 \end{aligned}$$

The verification is successful if the five verification formulas shown above all hold and fails if any one of the five formulas does not hold.

5 If the verification is successful (or the verification processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out four random numbers $s, t, k, h \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

10 The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' , C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{2F}' , C_{2F}'' , C_{5X} , C_{5Y} , C_{5Z} , C_{5FX} , C_{5FY} using these random numbers s, t, k, h , the ciphertext data C_i stored in the temporary data storage unit 42, the re-encryption key R_{ijL} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$\begin{aligned}
 C_{2X}' &= X_i^s, C_{2X}'' = C_{2X}^s = X_i^{rs}, C_{2Y}' = Y_{1i}^t, C_{2Y}'' = C_{2Y}^t = Y_{1i}^{rt}, \\
 C_{2Z}' &= Z_i^k, C_{2Z}'' = C_{2Z}^k = Z_i^{rk}, C_{2Z1}' = Z_{1i}^k, C_{2Z1}'' = C_{2Z1}^k = Z_{1i}^{rk}, \\
 C_{2F}' &= F_i(L)^h, C_{2F}'' = C_{2F}^h = F_i(L)^{rh},
 \end{aligned}$$

$$C_{5X} = R_{ijL1}^{\frac{1}{s}}, C_{5Y} = R_{ijL2}^{\frac{1}{t}}, C_{5Z} = R_{ijL3}^{\frac{1}{k}}, \\ C_{5FX} = R_{ijL4}^{\frac{1}{h}}, C_{5FY} = R_{ijL5}^{\frac{1}{h}}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data

5 $C_j = (L, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{2X},$
 $C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4, C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, i)$ by adding all the
above re-encrypted data C_{2X}' to C_{5FY} and the identification
information i of the re-encryption key generator 30 to the
ciphertext data C_i and writes the obtained ciphertext data
10 into the temporary data storage unit 42.

A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46 (ST43).

15 (5) Decryption processing

A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51, the public
20 parameter par in the temporary data storage unit 52, and the public key pk_i of the re-encryption key generator 30 identified by the identification information i of the re-encryption key generator 30 contained in the re-encrypted data C_j in the temporary data storage unit 52 based on the
25 following verification formulas (ST51).

$$\begin{aligned}
& e(C''_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(C'_{2X}, C_4), \\
& e(C''_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(C'_{2Y}, C_4), \\
& e(C''_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(C'_{2Z}, C_4), \\
& e(C''_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(C'_{2Z1}, C_4), \\
& e(C''_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(C'_{2F}, C_4), \\
& e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) = e(C_{5X}, C'_{2X}) \cdot e(g_2, g)^{y_j}, \\
& e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) = e(C_{5Y}, C'_{2Y}) \cdot e(g, g_4)^{x_j}, \\
& e(C_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(X_i, C_4), \\
& e(C_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(Y_{1i}, C_4), \\
& e(C_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(Z_i, C_4), \\
& e(C_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(Z_{1i}, C_4), \\
& e(C_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) = e(F_i(L), C_4). \dots (e3)
\end{aligned}$$

Instead of the seventh verification formula in the formula (e3), the following verification formula may be used.

$$5 \quad e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4).$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

10 The verification is successful if the 12 verification formulas in the formula (e3) all hold and fails if any one of the 12 formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as

shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

5

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z}) \cdot e(C_{5FX}, C''_{2F})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1}) \cdot e(C_{5FY}, C''_{2F})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

In the following calculation, g_1 is rewritten and denoted by g^α and g_2 is rewritten and denoted by g^β .

10

$$\begin{aligned} & C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z}) \cdot e(C_{5FX}, C''_{2F})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1}) \cdot e(C_{5FY}, C''_{2F})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\frac{\pi x_j + \beta y_j + \theta}{z_i k}}, g^{z_i r k}) \cdot e(g^{\frac{x_i \delta_x}{h}}, g^{(L+w_i)rh})}{e(g^{\frac{((\pi x_j + \theta)/x_i) + (L+w_i)\delta_x}{s}}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\pi x_j + \beta y_j + \theta}{z_i k}}, g^{\alpha z_i r k}) \cdot e(g^{\frac{\alpha y_i \delta_y}{h}}, g^{(L+w_i)rh})}{e(g^{\frac{((\beta y_j + \theta)/y_i) + (L+w_i)\delta_y}{t}}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\} \\ &= C_3 / \left\{ \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^r) \cdot e(g^{x_i \delta_x}, g^{(L+w_i)r})}{e(g^{\pi x_j + \theta + (L+w_i)x_i \delta_x}, g^r)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\pi x_j + \beta y_j + \theta}, g^{\alpha r}) \cdot e(g^{\alpha y_i \delta_y}, g^{(L+w_i)r})}{e(g^{\beta y_j + \theta + (L+w_i)y_i \delta_y}, g^{\alpha r})} \right)^{\frac{1}{x_j}} \right\} \\ &= m \cdot e(g_4 g_2, g)^r / (e(g, g)^{\beta r} \cdot e(g, g)^{\pi \alpha r}) = m. \end{aligned}$$

15

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ijL} based on the random numbers θ , δ_x , δ_y , the re-encryption key generation key rk_j , and the time parameter L , in addition to the effects of the ninth embodiment, whether to transfer the decryption right can be decided in each period so that finer access control can be exercised. Accordingly, even if the decryption right of

20

ciphertext for the transfer source (in the example of FIG. 8, the group administrator) is once transferred to the transfer destination (in the example of FIG. 8, for example, the user A) in a certain period, it is possible not to
5 grant the decryption right of the ciphertext for the transfer source to the relevant transfer destination, that is, the decryption right of the transfer destination (of the ciphertext for the relevant transfer source) can be invalidated in the next period and therefore, a file
10 sharing system whose convenience is further improved can be provided.

Also according to the present embodiment, as described above, due to the configuration of omitting the one-time signature σ in the fourth embodiment, the data size of
15 ciphertext data and re-encrypted data can be reduced when compared with the embodiment shown in the fourth embodiment. Also according to the present embodiment, the amount of computations needed for encryption processing and re-encryption processing can be reduced when compared with the
20 embodiment shown in the fourth embodiment.

[Twenty-first modification]

In the tenth embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption
25 device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is cited, but the tenth embodiment may be changed to a form in

which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present
5 modification is the same as the key setup processing in the tenth embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

The difference between the encryption processing in
10 the present modification and the encryption processing in the tenth embodiment is only in the last step. To provide a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case,
15 the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data
20 C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formulas (ST51').

$$\begin{aligned}
 e(C_{2X}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{x_i}, \\
 e(C_{2Y}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g_1, C_4)^{y_i}, \\
 e(C_{2Z}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{z_i}, \\
 e(C_{2Z1}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g_1, C_4)^{z_i}, \\
 e(C_{2F}, u^{H(L, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3)} \cdot v) &= e(g, C_4)^{L+w_i}.
 \end{aligned}$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

The verification is successful if the five verification formulas shown above all hold and fails if any one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(g_4 g_2, C_{2X})^{\frac{1}{x_i}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned}
 &m \cdot e(g_4 g_2, g)^r / e(g_4 g_2, C_{2X})^{\frac{1}{x_i}} \\
 &= m \cdot e(g_4 g_2, g)^r / e(g_4 g_2, g^{x_i r})^{\frac{1}{x_i}}
 \end{aligned}$$

$$= m \cdot e(g_4 g_2, g)^r / e(g_4 g_2, g)^r = m.$$

(Eleventh embodiment)

In the present embodiment, an example of omitting a one-time signature in the fifth embodiment will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

First, a public parameter generation unit 13 of a key generator 10 generates a public parameter

$\text{par} = \{\lambda, p, \mathbb{G}, \mathbb{G}_T, g, g_1, u, v, H\}$ based on a security parameter λ stored in a key parameter storage unit 11 in advance or acquires the parameter from outside (ST1). H is a hash function defined by the following formula.

$$H : \mathbb{G} \times \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$$

If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects an element x_i of $\mathbb{Z}_p^*$ from $\mathbb{Z}_p^*$ and sets a private key $\text{sk}_i = x_i$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $\text{pk}_i = X_i$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par .

$$X_i = g^{x_i}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i = X_{1i}$ using the private key sk_i and the public parameter par (ST3).

$$X_{1i} = g_1^{x_i}$$

5 Similarly, if the identification information of a decryption device 50 is assumed to be j , the key generation unit 14 randomly selects an element x_j of Z_p^* from Z_p^* and sets a private key $sk_j = x_j$ of the decryption device 50. Then, the key generation unit 14 generates a public key
10 $pk_j = X_j$ of the decryption device 50 using the private key sk_j and the public parameter par .

$$X_j = g^{x_j}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j = X_{1j}$ using the private key
15 sk_j and the public parameter par (ST6).

$$X_{1j} = g_1^{x_j}$$

(2) Encryption processing

No one-time signature is used in the present embodiment and therefore, processing corresponding to step
20 ST21 and step ST23 in the first embodiment is not performed.

An encryption parameter generation unit 23 of an encryption device 20 generates and sends out a random number $r \in Z_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted
25 data C_2, C_3, C_4 below from a message $m \in G_T$ as plaintext data

using the random number r , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST22).

$$C_2 = X_i^r, C_3 = e(g_1, g)^r \cdot m, C_4 = (u^{H(C_2, C_3)} \cdot v)^r$$

5 After step ST22 is completed, the ciphertext generation unit 24 generates ciphertext data $C_i = (C_2, C_3, C_4)$ including all the above encrypted data C_2 to C_4 and writes the obtained ciphertext data into a temporary data storage unit 21.

10 A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

15 Now, assume that the decryption device 50 (in the example of FIG. 8, for example, the user A) approves that ciphertext addressed to the re-encryption key generator 30 (in the example of FIG. 8, the group administrator) is re-encrypted into ciphertext addressed to the decryption device 50 (in the example of FIG. 8, for example, the user
20 A) (that is, the transfer of decryption right of the ciphertext to the user A). The decryption device 50 transmits the re-encryption key generation key rk_j of the decryption device 50 to the re-encryption key generator 30. A communication unit 33 of the re-encryption key generator
25 30 acquires and writes the re-encryption key generation key rk_j of the decryption device 50 into a temporary data storage unit 32 under the control of a control unit 35

(ST31).

Because there is no need to generate a random number for the generation of a re-encryption key in the present embodiment, a random number generation unit 36 of the re-encryption key generator 30 is not needed.

A re-encryption key generation unit 34 generates a re-encryption key R_{ij} as shown in the following formula using the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the re-encryption key generation key rk_j of the decryption device 50 in the temporary data storage unit 32, and the public parameter par (ST32).

$$R_{ij} = X_{1j}^{1/x_i} = g_1^{\frac{x_j}{x_i}}$$

(4) Re-encryption processing

A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par and the public key pk_i of the re-encryption key generator 30 based on the following verification formula (ST41).

$$e(C_2, u^{H(C_2, C_3)} \cdot v) = e(X_i, C_4).$$

The verification is successful if the above verification formula holds and fails if the above verification formula does not hold.

If the verification is successful (or the verification

processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out a random number $t \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_2' , C_2'' , C_2''' using the random number t , the ciphertext data C_i stored in the temporary data storage unit 42, the re-encryption key R_{ij} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$C_2' = X_i^t, C_2'' = R_{ij}^{1/t} = g_1^{\frac{x_j}{tx_i}}, C_2''' = C_2^t = X_i^{rt}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data $C_j = (C_2', C_2'', C_2''', C_2, C_3, C_4, i)$ by adding all the above re-encrypted data C_2' to C_2''' and the identification

information i of the re-encryption key generator 30 to the ciphertext data C_i and writes the obtained re-encrypted data C_j into the temporary data storage unit 42.

A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46 (ST43).

(5) Decryption processing

A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51, the public parameter par in the temporary data storage unit 52, and

the public key pk_i of the re-encryption key generator 30 identified by the identification information i of the re-encryption key generator 30 contained in the re-encrypted data C_j in the temporary data storage unit 52 based on the
5 following verification formulas (ST51).

$$\begin{aligned} e(C'_2, C''_2) &= e(g, g_1)^{x_j}, \\ e(C'''_2, u^{H(C_2, C_3)} \cdot v) &= e(C'_2, C_4), \\ e(C_2, u^{H(C_2, C_3)} \cdot v) &= e(X_i, C_4). \end{aligned} \quad \dots(f1)$$

Instead of the first verification formula in the formula (f1), the following verification formula may be used.

$$e(C'_2, C''_2) = e(X_j, g_1) \quad \dots(f2)$$

10

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

The verification is successful if the three
15 verification formulas in the formula (f1) all hold and fails if any one of the three formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as
20 shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit

52 to obtain the message m (ST52).

$$m = C_3 / e(C_2'', C_2''')^{1/x_j}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 / e(C_2'', C_2''')^{1/x_j} \\ &= C_3 / e(g_1^{\frac{x_j}{tx_i}}, g^{x_i rt})^{1/x_j} \\ &= m \cdot e(g_1, g)^r / e(g_1^{x_j}, g^r)^{1/x_j} = m. \end{aligned}$$

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ij} based on the re-encryption key generation key rk_j , effects similar to those in the fifth embodiment can be obtained.

Also according to the present embodiment, as described above, due to the configuration of omitting the one-time signature σ in the fifth embodiment, the data size of ciphertext data and re-encrypted data can be reduced when compared with the embodiment shown in the fifth embodiment. Also according to the present embodiment, the amount of computations needed for encryption processing and re-encryption processing can be reduced when compared with the embodiment shown in the fifth embodiment.

[Twenty-second modification]

In the eleventh embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption

device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is cited, but the eleventh embodiment may be changed to a form in which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present modification is the same as the key setup processing in the eleventh embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

The difference between the encryption processing in the present modification and the encryption processing in the eleventh embodiment is only in the last step. To provide a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formula (ST51').

$$e\left(C_2, u^{H(C_2, C_3)} \cdot v\right) = e(g, C_4)^{x_i} \quad \dots (f3)$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

5 The verification is successful if the verification formula in the formula (f3) holds and fails if the verification formula in the formula (f3) does not hold.

 If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device
10 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(C_2, g_1)^{1/x_i}$$

15 That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 / e(g_1, C_2)^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^r / e(g_1, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^r / e(g_1, g)^r = m. \end{aligned}$$

20 [Twenty-third modification]

 In addition to the twenty-second modification, as will be described below, the eleventh embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the
25 key setup processing, encryption processing, and decryption

processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the eleventh embodiment respectively.

5 Hereinafter, the encryption processing and the decryption processing in the present modification will be described. In the present modification, the identification information of the decryption device 50 is assumed to be j .

No one-time signature is used in the present
10 embodiment either and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed. The encryption parameter generation unit 23 of the encryption device 20 generates and sends out two random numbers $r, t \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

15 The ciphertext generation unit 24 generates encrypted data $C_2', C_2'', C_2''', C_2, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random numbers r, t , the public key pk_j of the decryption device 50, and the public parameter (ST22').

$$C_2' = X_j^t, C_2'' = g_1^{1/t}, C_2''' = X_j^{rt}, C_2 = X_j^r, \\ 20 \quad C_3 = e(g_1, g)^r \cdot m, C_4 = (u^{H(C_2, C_3)} \cdot v)^r$$

After step ST22' is completed, the ciphertext generation unit 24 generates ciphertext data $C_j = (C_2', C_2'', C_2''', C_2, C_3, C_4, j)$ including all the above encrypted data C_2' to C_4 and the identification information j of the decryption device 50 and writes the obtained
25 ciphertext data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

5 The decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in the temporary data storage unit 52 using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 10 52 based on the following verification formulas (ST51').

$$\begin{aligned} e(C'_2, C''_2) &= e(g, g_1)^{x_j}, \\ e(C'''_2, u^{H(C_2, C_3)} \cdot v) &= e(C'_2, C_4), \\ e(C_2, u^{H(C_2, C_3)} \cdot v) &= e(g, C_4)^{x_j}. \end{aligned} \quad \dots (f4)$$

Instead of the first verification formula in the formula (f4), the verification formula shown in the formula (f2) may be used. Also instead of the third verification 15 formula in the formula (f4), the following verification formula may be used.

$$e(C_2, u^{H(C_2, C_3)} \cdot v) = e(X_j, C_4).$$

The verification is successful if the three verification formulas in the formula (f4) all hold and 20 fails if any one of the three formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j of the local

device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(C_2'', C_2''')^{1/x_j}$$

5 That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 / e(C_2'', C_2''')^{1/x_j} \\ &= C_3 / e(g_1^{1/t}, g^{x_j r t})^{1/x_j} \\ &= m \cdot e(g_1, g)^r / e(g_1, g^r) = m. \end{aligned}$$

(Twelfth embodiment)

10 In the present embodiment, an example of updating a re-encryption key in the eleventh embodiment at certain intervals will be described. Hereinafter, key setup processing, encryption processing, re-encryption key generation processing, re-encryption processing, and
15 decryption processing in the present embodiment will be described. Portions common to the aforementioned first embodiment or second embodiment may or may not be described by using the same reference signs.

(1) Key setup processing

20 A public parameter generation unit 13 of a key generator 10 generates a public parameter par in the same manner as in the eleventh embodiment or acquires the parameter from outside (ST1). H is a hash function defined by the following formula.

$$H : \mathbb{Z}_p^* \times \mathbb{G}^2 \times \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$$

If the identification information of a re-encryption key generator 30 is i , a key generation unit 14 randomly selects elements x_i, y_i of Z_p^* from Z_p^* and sets a private key $sk_i=(x_i, y_i)$ of the re-encryption key generator 30. Then, the key generation unit 14 generates a public key $pk_i=(X_i, Y_i)$ of the re-encryption key generator 30 using the private key sk_i and the public parameter par .

$$X_i = g^{x_i}, Y_i = g^{y_i}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_i=X_{1i}$ using the private key sk_i and the public parameter par (ST3).

$$X_{1i} = g_1^{x_i}$$

Similarly, if the identification information of a decryption device 50 is assumed to be j , the key generation unit 14 randomly selects elements x_j, y_j of Z_p^* from Z_p^* and sets a private key $sk_j=(x_j, y_j)$ of the decryption device 50. Then, the key generation unit 14 generates a public key $pk_j=(X_j, Y_j)$ of the decryption device 50 using the private key sk_j and the public parameter par .

$$X_j = g^{x_j}, Y_j = g^{y_j}$$

Further, the key generation unit 14 generates a re-encryption key generation key $rk_j=X_{1j}$ using the private key sk_j and the public parameter par (ST6).

$$X_{1j} = g_1^{x_j}$$

(2) Encryption processing

No one-time signature is used in the present embodiment and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed.

5 An encryption parameter generation unit 23 of an encryption device 20 generates and sends out a random number $s \in \mathbb{Z}_p^*$ to a ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data C_1, C_2, C_3, C_4 below from a message $m \in G_T$ as plaintext data using the random number s , the public key pk_i of the re-encryption key generator 30, the public parameter par ,
10 and a time parameter L (ST22).

$$C_1 = X_i^s, C_2 = F_i(L)^s, C_3 = e(g_1, g)^s \cdot m, \\ C_4 = (u^{H(L, C_1, C_2, C_3)} \cdot v)^s$$

The function $F_i(L)$ is defined as shown below.

$$F_i(L) = g^L \cdot Y_i = g^{L+y_i} \quad (L \in \mathbb{Z}_p^*)$$

15 After step ST22 is completed, the ciphertext generation unit 24 generates ciphertext data $C_i = (L, C_1, C_2, C_3, C_4)$ including the time parameter L and all the above encrypted data C_1 to C_4 and writes the obtained ciphertext data into a temporary data storage unit 21.

20 A communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to a re-encryption device 40 under the control of a control unit 25 (ST24).

(3) Re-encryption key generation processing

Now, assume that the decryption device 50 (in the
25 example of FIG. 8, for example, the user A) approves that

ciphertext addressed to the re-encryption key generator 30
(in the example of FIG. 8, the group administrator) and
generated by using L as the time parameter value is re-
encrypted into ciphertext addressed to the decryption
5 device 50 (in the example of FIG. 8, for example, the user
A) (that is, the transfer of decryption right of the
ciphertext to the user A). The decryption device 50
transmits the re-encryption key generation key rk_j of the
decryption device 50 to the re-encryption key generator 30.
10 A communication unit 33 of the re-encryption key generator
30 acquires and writes the re-encryption key generation key
 rk_j of the decryption device 50 into a temporary data
storage unit 32 under the control of a control unit 35
(ST31).

15 A random number generation unit 36 generates and sends
out a random number $r \in \mathbb{Z}_p^*$ to a re-encryption key generation
unit 34.

The re-encryption key generation unit 34 generates a
re-encryption key R_{ijL} as shown in the following formulas
20 using the random number r , the private key sk_i of the re-
encryption key generator 30 in a private key storage unit
31, the re-encryption key generation key rk_j of the
decryption device 50 in the temporary data storage unit 32,
the public parameter par , and the time parameter L (ST32).

25 $R_{ijL} = (A_{ijL}, B_{ijL})$

$$A_{ijL} = (X_{1j}^{1/x_i} \cdot (g^{L+y_i})^r, B_{ijL} = g^{x_i r}$$

(4) Re-encryption processing

A re-encryption processing unit 44 of the re-encryption device 40 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter par , the time parameter L contained in the ciphertext data C_i in the temporary data storage unit 42, and the public key pk_i of the re-encryption key generator 30 based on the following verification formulas (ST41).

$$e(X_i, C_4) = e(C_1, u^{H(L, C_1, C_2, C_3)} \cdot v)$$
$$e(X_i, C_2) = e(C_1, F_i(L)).$$

The verification is successful if both verification formulas shown above hold and fails if any one of the two formulas does not hold.

If the verification is successful (or the verification processing in step ST41 is skipped), a re-encryption parameter generation unit 45 generates and sends out two random numbers $t_1, t_2 \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_1' , C_1'' , C_1''' , C_2' , C_2'' , C_2''' using the random numbers t_1 , t_2 , the ciphertext data C_i stored in the temporary data storage unit 42, the re-encryption key R_{ijL} , the public key pk_i of the re-encryption key generator 30, and the public parameter par (ST42).

$$C_1' = X_i^{t_1}, C_1'' = (A_{ij}L)^{1/t_1}, C_1''' = C_1^{t_1} = X_i^{st_1},$$
$$C_2' = F_i(L)^{t_2}, C_2'' = B_{ij}L^{1/t_2}, C_2''' = C_2^{t_2} = F_i(L)^{st_2}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data $C_j = (L, C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_1, C_2, C_3, C_4, i)$ by adding
5 all the above re-encrypted data C_1' to C_2''' and the identification information i of the re-encryption key generator 30 to the ciphertext data C_1 and writes the obtained ciphertext data into the temporary data storage unit 42.

10 A communication unit 43 transmits the re-encrypted data C_j in the temporary data storage unit 42 to the decryption device 50 under the control of a control unit 46 (ST43).

(5) Decryption processing

15 A decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the private key sk_j of the local device 50 in a private key storage unit 51, the public parameter par in the temporary data storage unit 52, and
20 the public key pk_i of the re-encryption key generator 30 identified by the identification information i of the re-encryption key generator 30 contained in the re-encrypted data C_j in the temporary data storage unit 52 based on the following verification formulas (ST51).

$$\begin{aligned} e(C_1''', u^{H(L, C_1, C_2, C_3)} \cdot v) &= e(C_1', C_4), \\ e(C_2''', u^{H(L, C_1, C_2, C_3)} \cdot v) &= e(C_2', C_4), \\ e(C_1', C_1'') &= e(g, g_1)^{x_j} \cdot e(C_2', C_2''), \\ e(X_i, C_4) &= e(C_1, u^{H(L, C_1, C_2, C_3)} \cdot v) \\ e(X_i, C_2) &= e(C_1, F_i(L)). \end{aligned} \quad \dots (f5)$$

Instead of the third verification formula in the formula (f5), the following verification formula may be used.

$$5 \quad e(C_1', C_1'') = e(X_j, g_1) \cdot e(C_2', C_2'') \quad \dots (f6)$$

In this case, the decryption processing unit 54 conducts verification using the public key pk_j of the local device 50.

10 The verification is successful if the five verification formulas in the formula (f5) all hold and fails if any one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST51 is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52).

$$15 \quad m = C_3 \cdot (e(C_2'', C_2''') / e(C_1'', C_1'''))^{1/x_j}.$$

20 That the decryption result of the above formula

produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 \cdot \{e(g^{x_i r/t_2}, F_i(L)^{st_2})/e((g_1^{x_j/x_i} \cdot F_i(L)^r)^{1/t_1}, g^{sx_i t_1})\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_i r}, F_i(L)^s)/e(g_1^{x_j/x_i} \cdot F_i(L)^r, g^{sx_i})\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_i r}, F_i(L)^s)/e(g_1^{x_j} \cdot F_i(L)^{x_i r}, g^s)\}^{1/x_j} \\ &= m \cdot e(g_1, g)^s \cdot \{1/e(g_1^{x_j}, g^s)\}^{1/x_j} = m. \end{aligned}$$

According to the present embodiment, as described
5 above, due to the configuration of generating the re-
encryption key R_{ijL} based on the random number r , the re-
encryption key generation key rk_j , and the time parameter L ,
in addition to the effects of the eleventh embodiment,
whether to transfer the decryption right can be decided in
10 each period so that finer access control can be exercised.
Accordingly, even if the decryption right of ciphertext for
the transfer source (in the example of FIG. 8, the group
administrator) is once transferred to the transfer
destination (in the example of FIG. 8, for example, the
15 user A) in a certain period, it is possible not to grant
the decryption right of the ciphertext for the transfer
source to the relevant transfer destination, that is, the
decryption right of the transfer destination (of the
ciphertext for the relevant transfer source) can be
20 invalidated in the next period and therefore, a file
sharing system whose convenience is further improved can be
provided.

Also according to the present embodiment, as described
above, due to the configuration of omitting the one-time

signature σ in the sixth embodiment, the data size of ciphertext data and re-encrypted data can be reduced when compared with the embodiment shown in the sixth embodiment. Also according to the present embodiment, the amount of computations needed for encryption processing and re-encryption processing can be reduced when compared with the embodiment shown in the sixth embodiment.

[Twenty-fourth modification]

In the twelfth embodiment, an example in which ciphertext data is generated by the encryption device 20, the ciphertext data is re-encrypted by the re-encryption device 40 to generate re-encrypted data, and the re-encrypted data is decrypted by the decryption device 50 is cited, but the twelfth embodiment may be changed to a form in which the ciphertext data is decrypted without being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present modification is the same as the key setup processing in the twelfth embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

The difference between the encryption processing in the present modification and the encryption processing in the twelfth embodiment is in the last step only. To provide a description by making use of the above symbols, the identification information of the decryption device 50

is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the private key sk_i of the local device 50 based on the following verification formula (ST51').

$$e(g, C_4)^{x_i} = e(C_1, u^{H(L, C_1, C_2, C_3)} \cdot v)$$
$$e(g, C_2)^{x_i} = e(C_1, g)^{L+y_i}. \quad \dots (f7)$$

Incidentally, the decryption device 50 may verify the ciphertext data C_i generated by the encryption device 20 using the public parameter par and the public key pk_i of the local device 50 in the same manner as in step ST41.

The verification is successful if both verification formulas in the formula (f7) hold and fails if any one of the two formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i of the local device 50 and the public parameter par to obtain the message m (ST52').

$$m = C_3 / e(C_1, g_1)^{\frac{1}{x_i}}$$

That the decryption result of the above formula

produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 / e(C_1, g_1)^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^s / e(g^{x_i s}, g_1)^{\frac{1}{x_i}} \\ &= m \cdot e(g_1, g)^s / e(g, g_1)^s = m. \end{aligned}$$

5 [Twenty-fifth modification]

In addition to the twenty-fourth modification, as will be described below, the twelfth embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the
10 key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the twelfth embodiment respectively.
15 Hereinafter, the encryption processing and the decryption processing in the present modification will be described. In the present modification, the identification information of the decryption device 50 is assumed to be j .

No one-time signature is used in the present
20 embodiment either and therefore, processing corresponding to step ST21 and step ST23 in the first embodiment is not performed. The encryption parameter generation unit 23 of the encryption device 20 generates and sends out two random numbers $t_1, t_2 \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

25 The ciphertext generation unit 24 generates encrypted data $C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_1, C_2, C_3, C_4$ below

from a message $m \in G_T$ as plaintext data using the random numbers t_1, t_2 , the public key pk_j of the decryption device 50, the public parameter, and a time parameter L (ST22').

$$\begin{aligned} C_1' &= X_j^{t_1}, C_1'' = (F_j(L) \cdot g_1)^{1/t_1}, C_1''' = X_j^{st_1}, \\ C_2' &= F_j(L)^{t_2}, C_2'' = X_j^{1/t_2}, C_2''' = F_j(L)^{st_2}, \\ C_1 &= X_j^s, C_2 = F_j(L)^s, C_3 = e(g_1, g)^s \cdot m, \\ C_4 &= (u^{H(L, C_1, C_2, C_3)} \cdot v)^s \end{aligned}$$

5 After step ST22' is completed, the ciphertext generation unit 24 generates ciphertext data $C_j = (L, C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_1, C_2, C_3, C_4, j)$ including the time parameter L , all the above encrypted data C_1' to C_4 , and the identification information j of the decryption device 50 and writes the obtained ciphertext data into the temporary data storage unit 21.

15 The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

20 The decryption processing unit 54 of the decryption device 50 verifies the re-encrypted data C_j in the temporary data storage unit 52 using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 based on the following verification formulas (ST51').

$$\begin{aligned}
 e(C_1''', u^{H(L, C_1, C_2, C_3)} \cdot v) &= e(C_1', C_4), \\
 e(C_2''', u^{H(L, C_1, C_2, C_3)} \cdot v) &= e(C_2', C_4), \\
 e(C_1', C_1'') &= e(g, g_1)^{x_j} \cdot e(C_2', C_2''), \\
 e(g, C_4)^{x_j} &= e(C_1, u^{H(L, C_1, C_2, C_3)} \cdot v) \\
 e(g, C_2)^{x_j} &= e(C_1, g)^{L+y_j}. \dots (f8)
 \end{aligned}$$

Instead of the third verification formula in the formula (f8), the verification formula shown in the formula (f6) may be used. Also instead of the fourth and fifth verification formulas in the formula (f8), the following verification formulas may be used.

$$\begin{aligned}
 e(X_j, C_4) &= e(C_1, u^{H(L, C_1, C_2, C_3)} \cdot v) \\
 e(X_j, C_2) &= e(C_1, F_j(L)).
 \end{aligned}$$

The verification is successful if the five verification formulas in the formula (f8) all hold and fails if any one of the five formulas does not hold.

If the verification is successful (or the verification processing in step ST51' is skipped), the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device 50 in the private key storage unit 51 and the public parameter par in the temporary data storage unit 52 to obtain the message m (ST52').

$$m = C_3 \cdot (e(C_2'', C_2''') / e(C_1'', C_1'''))^{1/x_j}.$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & C_3 \cdot \{e(C_2'', C_2''')/e(C_1'', C_1''')\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_j/t_2}, F_j(L)^{st_2})/e((F_j(L) \cdot g_1)^{1/t_1}, g^{sx_j t_1})\}^{1/x_j} \\ &= C_3 \cdot \{e(g^{x_j}, F_j(L)^s)/e(F_j(L) \cdot g_1, g^{sx_j})\}^{1/x_j} \\ &= m \cdot e(g_1, g)^s \cdot \{1/e(g_1, g^{sx_j})\}^{1/x_j} = m. \end{aligned}$$

According to at least one of the embodiments described above, due to the configuration of generating a re-encryption key based on a re-encryption key generation key, safe file sharing can be realized. For example, if the transfer destination (in the example of FIG. 8, for example, the user A) does not notify the transfer source (in the example of FIG. 8, the group administrator) of the re-encryption key generation key, a re-encryption key cannot be generated and therefore, re-encryption without approval of the transfer destination (that is, the transfer of decryption right of ciphertext to the relevant transfer destination) can be prevented.

Scenes of using each of the above embodiments will be described below.

The following three methods (1) to (3) can be cited in a file sharing system in which a user uploads a file to a server and the file is shared among a plurality of users as a method of maintaining confidentiality of file in the server.

(1) Individual key scheme in which a file is encrypted by an individual encryption key for each user

(2) Shared key scheme in which a file is encrypted by

an encryption key shared by each user

(3) Re-encryption scheme in which a file is encrypted using a proxy re-encryption scheme

5 In the above (1) to (3) schemes, the user who uploads a file to the server is assumed to be the user A and the user A shares the file with the user B and the user C.

10 In the individual key scheme in (1), each user has a set of a private key and a public key that differ from user to user. The user A encrypts a file by the public key of the user B (as an individual encryption key for the user B) and uploads the encrypted file to the server. Similarly, the user A encrypts the file by the public key of the user C (as an individual encryption key for the user C) and uploads the encrypted file to the server. That is, the
15 user A individually encrypts the file for each of the users who share the file.

In the shared key scheme in (2), each user shares a set of a private key and a public key common to each user. The user A encrypts a file by the public key (as an
20 encryption key common to each user) and uploads the encrypted file to the server. Each user shares the same private key.

In the proxy re-encryption scheme in (3), like the individual key scheme in (1), each user has a set of a
25 private key and a public key that differ from user to user. In contrast to the individual key scheme in (1), however, the user A only needs to encrypt a file by the public key

(hereinafter, called a group public key) of an entity
(hereinafter, called the group administrator) that manages
a group of users. The server uses the re-encryption key to
re-encrypt the encrypted file (uploaded by the user A) into
5 an encrypted file that can be decrypted only by each user.
Details of the proxy re-encryption scheme will be described
later.

In the individual key scheme in (1), if the file
should newly be shared also by a user D, a problem arises
10 that it is necessary for the user A to encrypt the file by
the public key of the user D (as an individual encryption
key of the user D) and upload the encrypted file to the
server. Therefore, the method in (1) complicates the
processing to add new users when the number of new users or
15 the number of shared file is large, making the method
unsuitable for a file sharing system.

In the shared key scheme in (2), a problem arises that
when a certain user (who has been allowed to share files
till a certain point in time) should be prevented from
20 being able to share files (the relevant user should be
excluded from the file sharing system) from that point in
time, a mechanism that updates the private key and public
key common to each user is separately needed or if the
private key common to each user should be leaked, all
25 encrypted files could be decrypted (by anyone who has
acquired the leaked key). Therefore, the shared key scheme
in (2) is unsuitable for a file sharing system.

In the proxy re-encryption scheme in (3), on the other hand, the server uses the re-encryption key to re-encrypt one piece of ciphertext into ciphertext that can be decrypted only by each user and thus, the above problems can be solved by adopting a configuration in which the user is not notified of the re-encryption key. Therefore, the proxy re-encryption scheme in (3) is suitable for a file sharing system. The proxy re-encryption scheme described above presents no particular problem, but if a problem should be brought up, a server is enabled to create re-encrypted text without permission of each user. Therefore, from the viewpoint of realizing safe file sharing, there is room for improvements of the proxy re-encryption scheme and the invention according to the present embodiment can realize safe file sharing.

Next, thirteenth and fourteenth embodiments using public information for re-encryption key generation will be described.

(Thirteenth embodiment)

The basic model of the proxy re-encryption scheme according to the thirteenth and fourteenth embodiments includes, like the above case, five functions. Similarly, depending on the implementation method of re-encryption, a model that differs in input into the functions and a model including other functions or keys than the above are also considered.

For example, a model, like the re-encryption scheme

shown in the present embodiment, in which the public key is divided into a public key for encryption and public information for generating a re-encryption key can be considered, but in conventional technology, a model in
5 which the division into the public key for encryption and public information for generating a re-encryption key is not implemented also exists.

Also, similarly to that described above, a model known as a non-interactive model and a model into which, instead
10 of a private key sk_A of the user A, a re-encryption key $rk_{A \rightarrow B}$ addressed to the user B and a private key sk_C of the user C are input are also considered.

Also, similarly to that described above, a model known as a unidirectional model and a model called a
15 bidirectional model are known.

Next, the configuration and effects of file sharing systems according to the thirteenth and fourteenth embodiments are similar to the configuration and effects of the file sharing system described using FIGS. 7 and 8. In
20 the thirteenth and fourteenth embodiments, however, instead of FIG. 8, FIG. 18 is used because public information for generating a re-encryption key is used.

FIG. 12 is a schematic diagram showing the configuration of a re-encryption system according to the
25 thirteenth embodiment. (The correspondence between FIGS. 12 and 18 will be described later.) The re-encryption system includes a key generator 10, an encryption device 20, a re-

encryption key generator 30, a re-encryption device 40, and
a decryption device 50. The key generator 10 generates
various parameters of the re-encryption system and also a
public key for encryption, public information for
5 generating a re-encryption key, and a private key for each
of the devices 40, 50.

The encryption device 20 transmits ciphertext data
obtained by encrypting plaintext data using a public key
for encryption corresponding to a private key of the re-
10 encryption key generator 30 to the re-encryption device 40.

The re-encryption key generator 30 generates a re-
encryption key using the private key of the re-encryption
key generator 30, the public information for generating a
re-encryption key of the decryption device 50, and a random
15 number.

The re-encryption device 40 transmits, to the
decryption device 50, re-encrypted data obtained by re-
encrypting the ciphertext data received from the encryption
device 20 by the re-encryption key without decrypting the
20 ciphertext data.

The decryption device 50 performs verification
processing of the re-encrypted data using the public key
for encryption of the re-encryption key generator 30 and
also performs decryption processing of the re-encrypted
25 data using the private key corresponding to the public key
for encryption of the local device 50 to obtain plaintext
data. While a plurality of units of the re-encryption key

generator 30, the re-encryption device 40, or the decryption device 50 may be provided, here a case in which one unit of each device is provided is taken as an example.

5 The correspondence between FIGS. 12 and 18 will be described. Each of the devices 10 to 50 is held by an entity that performs the relevant processing. When the user generates each of a public key for encryption, public information for generating a re-encryption key, and a private key of the user, the relevant user holds the key
10 generator 10.

If, for example, the user administrator or the group administrator generates each of a public key for encryption, public information for generating a re-encryption key, and a private key of the user, the user administrator or the
15 group administrator holds the key generator 10. The user holds the encryption device 20 or the decryption device 50 or both. The group administrator holds the re-encryption key generator 30.

If, for example, the user administrator or the server
20 Sv generates a re-encryption key, the user administrator or the server Sv holds the re-encryption key generator 30. The server Sv holds the re-encryption device 40. In the file sharing system illustrated in FIG. 18, the server Sv holds a public key storage unit 1, a ciphertext storage
25 unit 2, a re-encryption key storage unit 3, and a public information for generating re-encryption key storage unit 4 and the user C who performs encryption acquires a public

key for encryption used for encryption from the public key storage unit 1. However, the present embodiment is not limited to such an example and the user C may acquire, as will be described later, a public key for encryption used for encryption from the key generator 10 (that has generated the public key for encryption used for encryption). The server Sv stores ciphertext E (pkGr,m) generated by the user C in the ciphertext storage unit 2. The ciphertext storage unit 2 may be included in the re-encryption device 40 or an outside storage device not included in the re-encryption device 40.

Which entity holds which device is not limited to the above example and various variations can be considered. The user administrator or the group administrator may or may not serve also as a user. Each user device used by each user may be called a first user device, a second user device,... Similarly, the public key for encryption, the public information for generating a re-encryption key, and the private key generation key of each user device may be called the first public key for encryption, the first public information for generating a re-encryption key, and the first private key of the first user device, the second public key for encryption, the second public information for generating a re-encryption key, and the second private key of the second user device,...

Subsequently, the configuration of each of the devices 10 to 50 will concretely be described.

The key generator 10 includes a key generation parameter storage unit 11, a temporary data storage unit 12, a public parameter generation unit 13, a key generation unit 14, a communication unit 15, and a control unit 16.

5 The key generation parameter storage unit 11 is a storage device that stores key generation parameters.

The temporary data storage unit 12 is a storage device that stores temporary data such as data being processed by each of the generation units 13, 14 and processing results.

10 The public parameter generation unit 13 generates a public parameter for key generation.

The key generation unit 14 generates a public key for encryption, public information for generating a re-encryption key, and a private key addressed to a user.

15 The communication unit 15 is a communication interface to communicate between the key generator 10 and the other devices 20, 30, 40, 50 and has, for example, each of the following functions (f15-1), (f15-2):

(f15-1) Function to transmit the public key for encryption, public information for generating a re-encryption key, and private key of each of the devices 30, 50 in the temporary data storage unit 12 to each of the relevant devices 30, 50 under the control of the control unit 16.

25 (f15-2) Function to transmit the public key for encryption of the re-encryption key generator 30 in the temporary data storage unit 12 to the encryption device 20

under the control of the control unit 16.

In the description that follows, the description that transmission/reception goes through the communication unit 15 may be omitted to avoid a redundant description during transmission/reception. This also applies to each communication unit of the other devices 20, 30, 40, 50 than the key generator 10.

The control unit 16 has a function to control each of the units 11 to 15 of the key generator 10 such that each unit operates as shown in FIG. 13.

The encryption device 20 includes a temporary data storage unit 21, a communication unit 22, an encrypted data generation unit 23, a ciphertext generation unit 24, and a control unit 25.

The temporary data storage unit 21 is a storage device that stores a public key for encryption of the re-encryption key generator 30 received from the key generator 10 and transient data (hereinafter, written as "temporary data") such as data being processed by each of the generation devices 23, 24 and processing results.

The communication unit 22 is a communication interface to communicate between the encryption device 20 and the other devices 10, 30, 40, 50 and has, for example, each of the following functions (f22-1), (f22-2):

(f22-1) Function to acquire the public key for encryption of the re-encryption key generator 30 published to the key generator 10 and to write the public key for

encryption into the temporary data storage unit 21.

(f22-2) Function to transmit the ciphertext data in the temporary data storage unit 21 to the re-encryption device 40 under the control of the control unit 25.

5 The encryption parameter generation unit 23 has a function to generate an encryption parameter.

The ciphertext generation unit 24 has, for example, each of the following functions (f24-1), (f24-2):

(f24-1) Function to generate ciphertext data by
10 encrypting plaintext data using the public key for encryption (first public key of the first user device) of the re-encryption key generator 30 read from the temporary data storage unit 21.

(f24-2) Function to write the obtained ciphertext data
15 into the temporary data storage unit 21.

The control unit 25 has a function to control each of the units 21 to 24 such that each unit operates as shown in FIG. 14.

The re-encryption key generator 30 includes a private
20 key storage unit 31, a temporary data storage unit 32, a communication unit 33, a re-encryption key generation unit 34, a control unit 35, and a random number generation unit 36.

The private key storage unit 31 is a storage device
25 that stores a private key of the re-encryption key generator 30 received from the key generator 10.

The temporary data storage unit 32 is a storage device

that stores public information for generating a re-
encryption key of the decryption device 50 received from
the key generator 10 and temporary data such as data being
processed by the re-encryption device 34 and processing
5 results.

The communication unit 33 is a communication interface
to communicate between the re-encryption key generator 30
and the other devices 10, 20, 40, 50 and has, for example,
a function to transmit the re-encryption key in the
10 temporary data storage unit 32 to the re-encryption device
40 under the control of the control unit 35.

The re-encryption key generation unit 34 has, for
example, each of the following functions (f34-1), (f34-2):

(f34-1) Function to generate a re-encryption key based
15 on the private key of the re-encryption key generator 30
read from the private key storage unit 31 (first private
key of the first user device), the public information for
generating a re-encryption key read from the decryption
device 50 read from the temporary data storage unit 32
20 (second public information for generating a re-encryption
key of the second user device), and a random number
generated by the random number generation unit 36 (in this
function, a parameter, more specifically, a random number
for encryption is not needed).

25 The function (f34-1) to generate a re-encryption key
may generate a re-encryption key based on the first private
key, second public information for generating a re-

encryption key, and random number independently of
ciphertext data before re-encryption. The public key for
encryption of the re-encryption key generator 30 may be
generated based on the first private key and a plurality of
5 system fixed values. The second public information for
generating a re-encryption key may be generated based on
the second private key and a plurality of system fixed
values.

(f34-2) Function to write the re-encryption key into
10 the temporary data storage unit 32.

The control unit 35 has a function to control each of
the units 31 to 34, 36 such that each unit operates as
shown in FIG. 15.

The random number generation unit 36 has a function to
15 generate and send out a random number to the re-encryption
key generation unit 34.

The re-encryption device 40 includes a re-encryption
key storage unit 41, a temporary data storage unit 42, a
communication unit 43, a re-encryption processing unit 44,
20 a re-encryption parameter generation unit 45, and a control
unit 46.

The re-encryption key storage unit 41 is a storage
unit that stores the re-encryption key received from the
re-encryption key generator 30.

25 The temporary data storage unit 42 is a storage device
that stores temporary data such as data being processed by
the re-encryption processing unit 44 and processing results.

The communication unit 43 is a communication interface to communicate between the re-encryption device 40 and the other devices 10, 20, 30, 50 and has, for example, each of the following functions (f43-1), (f43-2):

5 (f43-1) Function to send out ciphertext data received from the encryption device 20 to the re-encryption processing unit 44.

 (f43-2) Function to transmit the re-encrypted data in the temporary data storage unit 42 to the decryption device
10 50 under the control of the control unit 46.

The re-encryption processing unit 44 has, for example, each of the following functions (f44-1), (f44-2):

 (f44-1) Function to obtain re-encrypted data by, when ciphertext data is received from the encryption device 20,
15 re-encrypting the ciphertext data using the re-encryption key read from the re-encryption key storage unit 41 without decrypting the ciphertext data.

 (f44-2) Function to write the obtained re-encrypted data into the temporary data storage unit 42.

20 The control unit 46 has a function to control each of the units 41 to 45 such that each unit performs an operation (not shown) to deliver a re-encrypted text verification program and operates as shown in FIG. 16.

 The decryption device 50 includes a private key
25 storage unit 51, a temporary data storage unit 52, a communication unit 53, a decryption processing unit 54, and a control unit 56.

The private key storage unit 51 is a storage device that stores a private key of the decryption device 50 received from the key generator 10.

5 The temporary data storage unit 52 is a storage device that stores a public key for encryption/public information for generating a re-encryption key of the decryption device 50 received from the key generator 10 and a public key for encryption of the re-encryption key generator 30 and also temporary data such as data being processed by the
10 decryption processing unit 54 and processing results.

The communication unit 53 is a communication interface to communicate between the decryption device 50 and the other devices 10, 20, 30, 40 and has, for example, each of the following functions (f53-1) to (f53-3):

15 (f53-1) Function to write a private key of the decryption device 50 received from the key generator 10 into the private key storage unit 51.

(f53-2) Function to write a public key for encryption/public information for generating a re-
20 encryption key of the decryption device 50 received from the key generator 10 and a public key for encryption of the re-encryption key generator 30 into the temporary data storage unit 52.

(f53-3) Function to send out re-encrypted data
25 received from the re-encryption device 40 to the decryption processing unit 54.

The decryption processing unit 54 has, for example,

each of the following functions (f54-1) to (f54-3):

(f54-1) Function to obtain plaintext data by, when re-encrypted data is received from the re-encryption device 40, performing decryption processing on the re-encrypted data based on the private key of the decryption device 50 read from the private key storage unit 51 (second private key of the second user device).

(f54-2) Function to write the obtained plaintext data into the temporary data storage unit 52.

10 The control unit 56 has a function to control each of the units 51 to 54 such that each unit operates as shown in FIG. 17.

Next, the operation of the re-encryption system configured as described above will be described using the sequence diagrams of FIGS. 13 to 17.

15 The following operation takes a case in which processing is performed in a procedure of (1) key setup processing, (2) encryption processing, (3) re-encryption key generation processing, (4) re-encryption processing, and (5) decryption processing as an example. However, in the following operation, processing is not necessarily performed following the above procedure. For example, processing of re-encryption key generation may be performed before the encryption processing. Also, ciphertext data may be decrypted without the re-encryption processing being performed.

(1) The key setup processing is performed by the key

generator 10 as shown in FIG. 13 and each of steps ST1 to ST12 below.

First, the public parameter generation unit 13 of the key generator 10 generates a public parameter

5 $(p, \lambda, G, G_T, g, g_1, g_2, g_3, g_4, u, v, \text{Sig}(\hat{G}, \hat{S}, \hat{V}))$ or acquires the public parameter from outside (ST1).

More specifically, the public parameter generation unit 13 generates bilinear map groups (G, G_T) satisfying the order $p > 2\lambda$, elements $g, g_1, g_2, g_3, g_4, u, v$ of G , and a one-time signature algorithm $\text{Sig}(\hat{G}, \hat{S}, \hat{V})$ satisfying strong
10 unforgeability based on a security parameter λ stored in the key parameter storage unit 11 in advance.

Z_p^* is a set $(= (Z/pZ)^*)$ of integers mutually prime to Z_p and p and may also be called a multiplicative group Z_p^*
15 for the prime number p . Z_p is a set $(= (Z/pZ))$ of integers equal to 0 or greater and less than p . The "element" is a term having a mathematical meaning, is also called a constituent, and refers to, in a set constituted of a plurality of "things", individual things constituting the
20 set.

More specifically, the public parameter generation unit 13 selects bilinear map groups (G, G_T) satisfying the order $p > 2\lambda$ based on the security parameter λ stored in the key generation parameter storage unit 11 in advance. In
25 this case, p is an integer. p may be a prime number or a composite number. The public parameter generation unit 13 randomly selects elements g, g_1, g_2, u, v of G from G . The

public parameter generation unit 13 randomly selects ξ from Z_p^* and calculates $g_3=g\xi$ and $g_4=g_1\xi$. Z_p^* is a set $(=(Z/pZ)^*)$ of integers mutually prime to Z_p and p and may also be called a multiplicative group Z_p^* for p . Z_p is a set
5 $(=(Z/pZ)$ of integers equal to 0 or greater and less than p . g_3 is a value based on a random number ξ and a system fixed value g . g_4 is a value based on the random number ξ and a system fixed value g_1 .

$\hat{G}$ in the one-time signature algorithm $\text{Sig}(\hat{G}, \hat{S}, \hat{V})$
10 (hereinafter, $\text{Sig}(\hat{G}, \hat{S}, \hat{V})$ is also called Sig) means a function to generate a one-time key pair (ssk, svk) , $\hat{S}$ means a function to generate a signature σ for a message M , and $\hat{V}$ means a function to validate the signature σ .

For more information about the one-time signature, see
15 A. Menezes, P. van Oorschot, S. Vanstone, "Handbook of Applied Cryptography", CRC Press, (1996) pp.462-471, (1996).

The bilinear map (denoted by e) is a map yielding $e:G \times G \rightarrow G_T$ and satisfies the following three properties:

1. $e(ga, hb) = e(g, h)^{ab}$ holds for any $(g, h) \in G \times G$ and
20 $a, b \in Z$, where Z is a set of integers.

2. $e(g, h)$ is computable for any $(g, h) \in G \times G$.

3. $e(g, h) \neq 1_{G_T}$ always holds if $g, h \neq 1_G$, where 1_G is the unit element of G and 1_{G_T} is the unit element of G_T .

The bilinear map group (denoted by G, G_T) is a group
25 of the prime number order p for which a bilinear map $e:G \times G \rightarrow G_T$ exists. The above definition also means that if $g_1 = g^\alpha$ and $g_2 = g^\beta$ for elements g, g_1, g_2 of G , the following

equation holds (α and β are the exponents relating to elements g, g_1, g_2 of G).

$$e(g_1g_2, g) = e(g, g_1g_2) = e(g, g^\alpha \cdot g^\beta) = e(g, g^{\alpha+\beta}) = e(g, g)^\alpha + \beta = e(g, g)^\alpha \cdot e(g, g)^\beta$$

5 Incidentally, a plurality of elements g, g_1, g_2 of G is a plurality of preset system fixed values. The plurality of system fixed values is not limited to three elements of the bilinear map group G and a plurality of elements such as two or four or more elements of G can also
10 be used when appropriate. The term of "system fixed value" may be changed to a "fixed value", an "element", or a "system parameter". The plurality of system fixed values are a plurality of elements of a bilinear map group as a group of prime number orders for which a bilinear map
15 exists.

 In this specification, the notation assuming that both G and G_T are multiplicative groups is adopted, but the embodiments are not limited to such a notation and each of G, G_T can be represented by a notation assuming that each
20 is an additive group. That is, for example, G may be represented as an additive group and G_T as a multiplicative group. As the bilinear map, a map that yields $e: G_1 \times G_2 \rightarrow G_T$ for bilinear map groups G_1, G_2, G_T (G_1 and G_2 are different groups) may also be used. In this case, each element of G
25 in each piece of processing described later may be rewritten to an element of G_1 or an element of G_2 when appropriate. The above also apply to other embodiments.

Subsequently, the public parameter generation unit 13 writes the generated public parameter par into the temporary data storage unit 12. The key generator 10 publishes the public parameters

5 $p, \lambda, G, G_T, g, g_1, g_2, g_3, g_4, u, v, \text{Sig})$ in the temporary data storage unit 12 (ST2). If a public parameter is published before step ST1 is executed, step ST1 and step ST2 may be skipped by writing the public parameter into the temporary data storage unit 12.

10 If the identification information of the re-encryption key generator 30 is i , the key generation unit 14 generates a private key $x_i, y_i, z_i \in \mathbb{Z}_p^*$ of the re-encryption key generator 30 and generates a public key for encryption of the re-encryption key generator 30 using the private key
15 $\text{sk}_i = (x_i, y_i, z_i)$ (ST7).

$$pk_i = (X_i, Y_i, Y_{1i}, Z_i, Z_{1i})$$

$$X_i = g^{x_i}, Y_i = g^{y_i}, Y_{1i} = g_1^{y_i}, Z_i = g^{z_i}, Z_{1i} = g_1^{z_i}$$

If a twenty-seventh modification described later is not made, data Y_i may be omitted. That is, the public key
20 pk_i only needs to include data X_i, Y_{1i}, Z_i, Z_{1i} .

Also, the key generation unit 14 generates public
information $pi_i = (X_{3i}, Y_{2i})$ for

generating a re-encryption key of the re-encryption key generator 30 using the private key $\text{sk}_i = (x_i, y_i, z_i)$ (ST7').

Here,

$$X_{3i} = g_3^{x_i}, Y_{2i} = g_2^{y_i}$$

Subsequently, the key generation unit 14 writes the generated public key for encryption, public information for generating a re-encryption key, and private key into the temporary data storage unit 12. The communication unit 15 transmits the private key sk_i in the temporary data storage unit 12 to the re-encryption key generator 30 under the control of the control unit 16 (ST8). The key generator 10 publishes the public key for encryption pk_i of the re-encryption key generator 30 in the temporary data storage unit 12 (ST9).

Similarly, if the identification information of the decryption device 50 is j , the key generation unit 14 generates a private key $sk_j = (x_j, y_j, z_j)$ of the decryption device 50 and generates a public key for encryption of the decryption device 50 using the private key sk_j (ST10).

$$pk_j = (X_j, Y_j, Y_{1j}, Z_j, Z_{1j})$$

Here,

$$X_j = g^{x_j}, Y_j = g^{y_j}, Y_{1j} = g_1^{y_j}, Z_j = g^{z_j}, Z_{1j} = g_1^{z_j}$$

Similarly as described above, if the twenty-seventh modification described later is not made, data Y_j may be omitted. That is, the public key pk_j only needs to include data X_j, Y_{1j}, Z_j, Z_{1j} .

Also, the key generation unit 14 generates public information for generating a re-encryption key of the decryption device 50 using the private key sk_j (ST10').

$$pi_j = (X_{3j}, Y_{2j})$$
$$X_{3j} = g_3^{x_j}, Y_{2j} = g_2^{y_j}$$

5

Subsequently, the key generation unit 14 writes the generated public key for encryption, public information for generating a re-encryption key, and private key into the temporary data storage unit 12. The communication unit 15 transmits the private key sk_j in the temporary data storage unit 12 to the decryption device 50 under the control of the control unit 16 (ST11). The key generator 10 publishes the public information for generating a re-encryption key pi_j of the decryption device 50 in the temporary data storage unit 12 (ST12). If desired, regarding the private key sk_h , the public key for encryption pk_h , and the public information for generating a re-encryption key pi_h , the public key for encryption pk_h of the encryption device 20, the private key sk_h may be transmitted to the encryption device 20 to publish the public key for encryption pk_h and the public information for generating a re-encryption key pi_h in accordance with purposes by performing processing similar to each of steps ST10 to ST12.

With the above processing, the key setup processing is completed. Hereinafter, each of the devices 20, 30, 40, 50

25

can acquire and use the public parameter or public key published in steps ST2, ST6, ST9, ST12 when appropriate.

(2) The encryption processing is performed by the encryption device 20 as shown in FIG. 14 and each of steps ST21 to ST24 below.

The encryption parameter generation unit 23 of the encryption device 20 generates a key pair $(ssk, svk) = \hat{G}(\lambda)$ of a signature key ssk and a verification key svk in a one-time signature using the security parameter λ and a key pair generation function $\hat{G}$ in the public parameter par (ST21) and sets the verification key svk to encrypted data C_1 ($C_1 = svk$).

The encryption parameter generation unit 23 also generates and sends out a random number $r \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random number r and the public key for encryption pk_i of the re-encryption key generator 30 (ST22).

$$C_{2X} = X_i^r, C_{2Y} = Y_{li}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{li}^r,$$
$$C_3 = e(g_2 g_4, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

After step ST22 is completed, the ciphertext generation unit 24 generates a one-time signature σ as shown in the following formula for the encrypted data C_3 ,

C_4 based on the signature generation function $\hat{S}$ in the public parameter and the signature key ssk generated in step ST21 (ST23).

$$\sigma = \hat{S}(ssk, (C_3, C_4))$$

5 Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_i = (C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, \sigma)$ including all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext data into the temporary data storage unit 21. If the
10 verification is omitted, the ciphertext data $C_i = (C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3, C_4, \sigma)$ may be set as $C_i = (C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)$ by omitting verification data (C_1, C_4, σ) that is not used for decryption. In this case, processing to generate the verification data (C_1, C_4, σ) is
15 also omitted.

 In any case, the communication unit 22 transmits the ciphertext data C_i in the temporary data storage unit 21 to the re-encryption device 40 under the control of the control unit 25 (ST24).

20 With the above processing, the encryption processing is completed.

 In the present embodiment, as described above (also, as will be described later), verification may be omitted or generation of verification data may be omitted and this
25 also applies to an embodiment and modifications described later.

(3) The re-encryption key generation processing is

performed by the re-encryption key generator 30 as shown in FIG. 15 and each of steps ST31 to ST33 below.

That is, the communication unit 33 of the re-encryption key generator 30 acquires and writes the public information for generating a re-encryption key pi_j of the decryption device 50 published by the key generation unit 10 into the temporary data storage unit 32 under the control of the control unit 35 (ST31). In step ST5 described above, the communication unit 33 has received the private key sk_i of the re-encryption key generator 30 from the key generator 10 and written the private key into the private key storage unit 31,

The random number generation unit 36 generates and sends out a random number $\gamma \in Z_p^*$ to the re-encryption key generation unit 34.

The re-encryption key generation unit 34 generates a re-encryption key R_{ij} as shown in the following formulas based on the random number γ , the private key sk_i of the re-encryption key generator 30 in the private key storage unit 31, and the public information for generating a re-encryption key pi_j of the decryption device 50 in the temporary data storage unit 32 (ST32).

$$R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$$

$$R_{ij1} = (X_{3j} \cdot g^{\gamma})^{\frac{1}{x_i}} = g^{\frac{\xi x_j + \gamma}{x_i}},$$

$$R_{ij2} = (Y_{2j} \cdot g^{\gamma})^{\frac{1}{y_i}} = g^{\frac{\beta y_j + \gamma}{y_i}},$$

$$R_{ij3} = (X_{3j} \cdot Y_{2j} \cdot g^{\gamma})^{\frac{1}{z_i}} = g^{\frac{\xi x_j + \beta y_j + \gamma}{z_i}}$$

In the above formula, g_1 is rewritten and denoted by g^{α} , g_2 is rewritten and denoted by g^{β} , g_3 is rewritten and denoted by g^{ξ} , and g_4 is rewritten and denoted by $g^{\alpha\xi}$.

5 Thereafter, the re-encryption key generation unit 34 writes the generated re-encryption key R_{ij} into the temporary data storage unit 32. The communication unit 33 transmits the re-encryption key R_{ij} in the temporary data storage unit 32 to the re-encryption device 40 under the
10 control of a control unit 35 (ST33).

With the above processing, the re-encryption key generation processing is completed.

(4) The re-encryption processing is performed by the re-encryption device 40 as shown in FIG. 16 and each of
15 steps ST41 to ST43 below.

The communication unit 43 of the re-encryption device 40 writes the ciphertext data C_i transmitted in step ST24 and the re-encryption key R_{ij} transmitted in step ST33 into

the temporary data storage unit 42.

The re-encryption processing unit 44 verifies the ciphertext data C_i in the temporary data storage unit 42 using the public parameter and the following verification formula (ST41).

5

$$\begin{aligned}e(C_{2X}, u^{C_1} \cdot v) &= e(X_i, C_4), \\e(C_{2Y}, u^{C_1} \cdot v) &= e(Y_{1i}, C_4), \\e(C_{2Z}, u^{C_1} \cdot v) &= e(Z_i, C_4), \\e(C_{2Z1}, u^{C_1} \cdot v) &= e(Z_{1i}, C_4), \\ \hat{V}(C_1, \sigma, (C_3, C_4)) &= 1.\end{aligned}$$

The verification is successful if the five verification formulas shown above all hold and fails if any one of the five formulas does not hold.

10

If the verification is successful, the re-encryption parameter generation unit 45 generates and sends out three random numbers $s, t, k \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

15

The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' , C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{5X} , C_{5Y} , C_{5Z} using these random numbers s, t, k , the ciphertext data C_i in the temporary data storage unit 42, and the re-encryption key R_{ij} in the temporary data storage unit 42 (ST42).

$$\begin{aligned} C'_{2X} &= X_i^s, C''_{2X} = C_{2X}^s = X_i^{rs}, C'_{2Y} = Y_{1i}^t, C''_{2Y} = C_{2Y}^t = Y_{1i}^{rt}, \\ C'_{2Z} &= Z_i^k, C''_{2Z} = C_{2Z}^k = Z_i^{rk}, C'_{2Z1} = Z_{1i}^k, C''_{2Z1} = C_{2Z1}^k = Z_{1i}^{rk}, \\ C_{5X} &= R_{ij1}^{\frac{1}{s}}, C_{5Y} = R_{ij2}^{\frac{1}{t}}, C_{5Z} = R_{ij3}^{\frac{1}{k}} \end{aligned}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data

5 $C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X},$
 $C_{5Y}, C_{5Z}, C_3, C_4, \sigma)$ by replacing the encrypted data $C_{2X}, C_{2Y}, C_{2Z},$
 C_{2Z1} in the ciphertext data C_i by all the above encrypted
data/re-encrypted data C_{2X}' to C_{5Z} and writes the obtained
re-encrypted data C_j into the temporary data storage unit
10 42. If the verification is omitted, the re-encrypted data
 $C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X},$
 $C_{5Y}, C_{5Z}, C_3, C_4, \sigma)$ may be set as $C_j = (C_{2X}'', C_{2Y}'', C_{2Z}'', C_{2Z1}'', C_{5X},$
 $C_{5Y}, C_{5Z}, C_3)$ by omitting verification data
 $(C_1, C_{2X}', C_{2Y}', C_{2Z}', C_{2Z1}', C_4, \sigma)$ that is not used for decryption.
15 In this case, processing to generate the verification data
 $(C_{2X}', C_{2Y}', C_{2Z}', C_{2Z1}')$ is also omitted.

In any case, the communication unit 43 transmits the
re-encrypted data C_j in the temporary data storage unit 42
to the decryption device 50 under the control of the
20 control unit 46 (ST43).

With the above processing, the re-encryption processing is completed.

(5) The decryption processing is performed by the
decryption device 50 as shown in FIG. 17 and each of steps
25 ST51 and ST52 below.

That is, the communication unit 53 of the decryption device 50 receives the re-encrypted data C_j transmitted in step ST43 and writes the re-encrypted data C_j into the temporary data storage unit 52.

5 The decryption processing unit 54 verifies the re-encrypted data C_j in the temporary data storage unit 52 using the public parameter, the public key for encryption pk_j of the local device 50, and the following verification formulas (ST51).

$$e(C''_{2X}, u^{C_1} \cdot v) = e(C'_{2X}, C_4)$$

$$e(C''_{2Y}, u^{C_1} \cdot v) = e(C'_{2Y}, C_4)$$

$$e(C''_{2Z}, u^{C_1} \cdot v) = e(C'_{2Z}, C_4)$$

$$e(C''_{2Z1}, u^{C_1} \cdot v) = e(C'_{2Z1}, C_4)$$

$$e(C_{5Z}, C'_{2Z}) = e(C_{5X}, C'_{2X}) \cdot e(Y_j, g_2)$$

$$e(C_{5Z}, C'_{2Z1}) = e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4)$$

10 $\hat{V}(C_1, \sigma, (C_3, C_4)) = 1$

The verification is successful if the seven verification formulas shown above all hold and fails if any one of the seven formulas does not hold.

15 If the verification is successful, the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation. In the following calculation, g_1 is rewritten and denoted by g^α , g_2 is rewritten and denoted by g^β , g_3 is rewritten and denoted by g^ξ , and g_4 is rewritten and denoted by $g^{\alpha\xi}$.

$$\begin{aligned} & \frac{C_3}{\left\{ \left(\frac{e(C_{5Z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}} \\ &= \frac{C_3}{\left\{ \left(\frac{e(g^{\frac{\xi x_j + \beta y_j + \gamma}{z_i k}}, g^{z_i r k})}{e(g^{\frac{\xi x_j + \gamma}{x_i s}}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\xi x_j + \beta y_j + \gamma}{z_i k}}, g^{\alpha z_i r k})}{e(g^{\frac{\beta y_j + \gamma}{y_i t}}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\}} \\ &= \frac{m \cdot e(g_2 g_4, g)^r}{(e(g, g)^{\beta r} \cdot e(g, g)^{\alpha \xi r})} = m \end{aligned}$$

Incidentally, the decryption processing unit 54 may perform the decryption processing in step ST52 by omitting the verification processing in step ST51. Also, when no verification data is contained in the re-encrypted data C_j , the decryption processing unit 54 performs the decryption processing in step ST52 by omitting the verification processing in step ST51. The above m may be a key, instead

of a message itself. For example, a shared key in shared key cryptography may be set as $m=key$ to replace m with key. In this case, ciphertext obtained by encrypting a message using the shared key may be added to the ciphertext data or
5 re-encrypted data. The above also apply to the embodiment and modifications described later.

In the present embodiment, the order of processing may be changed when appropriate. For example, the order of the decryption processing and the ciphertext verification
10 processing may be changed. Similarly, because a parameter (more specifically, a random number) for performing encryption processing is not needed in the re-encryption key generation processing, the order of the re-encryption key generation processing may be changed to before the
15 encryption processing.

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ij} based on the random number γ , even if a server and a user collude with each other, the decryption
20 right can be prevented from being re-transferred without permission of the transfer source. Accordingly, there is no need to demand extreme reliability from the server and a file sharing system the user can safely use can be provided.

To complement the above description, there has been a
25 problem that, if the server operator, the user B, and the user E collude, a third party can calculate a normal re-encryption key ($rk_{A \rightarrow E}$) to re-encrypt ciphertext for the

user A into ciphertext for the user E by using a re-
encryption key ($rk_{A \rightarrow B}$) to re-encrypt the ciphertext for the
user A into ciphertext for the user B, a private key (sk_B)
of the user B, and a private key (sk_E) of the user E
5 without permission of the user A as the transfer source.

According to the present embodiment, however, due to
the configuration of generating the re-encryption key R_{ij}
based on the random number γ , even if the server operator,
the user B, and the user E collude, the decryption right
10 can be prevented from being re-transferred without
permission of the transfer source.

Also, due to the configuration of generating a re-
encryption key using public information for generating a
re-encryption key of the original transfer destination, no
15 re-encryption key can be generated without the public
information for generating a re-encryption key being
provided by the transfer destination. Therefore, a re-
encryption key can be generated with the permission of the
original transfer destination.

20 [Twenty-sixth modification]

In the thirteenth embodiment, an example in which
ciphertext data is generated by the encryption device 20,
the ciphertext data is re-encrypted by the re-encryption
device 40 to generate re-encrypted data, and the re-
25 encrypted data is decrypted by the decryption device 50 is
cited. However, the thirteenth embodiment may be changed
to a form in which the ciphertext data is decrypted without

being re-encrypted. In this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing in the present modification is the same as the key setup processing in the thirteenth embodiment. Hereinafter, the encryption processing and the decryption processing in the present modification will be described.

The difference between the encryption processing in the present modification and the encryption processing in the thirteenth embodiment is only in the last step. To provide a description by making use of the above symbols, the identification information of the decryption device 50 is assumed to be i for the sake of convenience. In this case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 in the same manner as in step ST41. If the verification is successful, the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i to obtain the message m .

$$m = \frac{C_3}{e(g_2 g_4, C_{2X})^{\frac{1}{x_i}}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_2 g_4, g)^r \bigg/ e(g_2 g_4, C_{2X})^{\frac{1}{x_i}} \\ &= m \cdot e(g_2 g_4, g)^r \bigg/ e(g_2 g_4, g^{x_i r})^{\frac{1}{x_i}} \\ &= m \cdot e(g_2 g_4, g)^r \bigg/ e(g_2 g_4, g)^r = m \end{aligned}$$

5 [Twenty-seventh modification]

In addition to the twenty-sixth modification, as will be described below, the thirteenth embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the thirteenth embodiment respectively.

10

Hereinafter, the encryption processing and the decryption processing in the present modification will be described. In the present modification, the identification information of the decryption device 50 is assumed to be j .

5 The encryption parameter generation unit 23 of the encryption device 20 generates (ssk, svk) in the same manner as in step ST21 (ST21') and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

10 The encryption parameter generation unit 23 also generates and sends out five random numbers $r, s, t, k, \gamma \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

 The ciphertext generation unit 24 generates encrypted data $C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random numbers r, s, t, k, γ and the public key for encryption pk_j of the decryption device 50 (ST22').

$$\begin{aligned} C'_{2X} &= Y_j^s, C''_{2X} = Y_j^{rs}, C'_{2Y} = X_j^t, C''_{2Y} = X_j^{rt}, \\ C'_{2Z} &= Y_j^k, C''_{2Z} = Y_j^{rk}, C'_{2Z1} = X_j^k, C''_{2Z1} = X_j^{rk}, \\ C_3 &= e(g_2 g_4, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r, \\ C_{5X} &= (g_4 \cdot g^\gamma)^{\frac{1}{s}}, C_{5Y} = (g_2 \cdot g^\gamma)^{\frac{1}{t}}, C_{5Z} = (g_4 \cdot g_2 \cdot g^\gamma)^{\frac{1}{k}} \end{aligned}$$

20 After step ST22' is completed, the ciphertext generation unit 24 generates a one-time signature σ in the same manner as in step ST23.

 Thereafter, the ciphertext generation unit 24

generates ciphertext data $C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{5X}, C_{5Y}, C_{5Z}, C_3, C_4, \sigma)$ including all the above encrypted data C_1 to C_4 and the one-time signature σ and writes the obtained ciphertext data into the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25.

The decryption device 50 verifies the ciphertext data C_j generated by the encryption device 20 in the same manner as in step ST51. If the verification is successful, the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j to obtain the message m .

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation. In the following calculation, g_1 is rewritten and denoted by g^α , g_2 is rewritten and denoted by g^β , g_3 is rewritten and denoted by g^ξ , and g_4 is rewritten and denoted by $g^{\alpha\xi}$.

$$\begin{aligned}
 & \frac{C_3}{\left\{ \left(\frac{e(C_{5z}, C''_{2Z})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5z}, C''_{2Z1})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{C_3}{\left\{ \left(\frac{e(g^{\frac{\alpha\xi+\beta+\gamma}{k}}, g^{y_jrk})}{e(g^{\frac{\alpha\xi+\gamma}{s}}, g^{y_jrs})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\alpha\xi+\beta+\gamma}{k}}, g^{x_jrk})}{e(g^{\frac{\beta+\gamma}{t}}, g^{x_jrt})} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{C_3}{\left\{ \left(\frac{e(g^{(\alpha\xi+\beta+\gamma)y_jr}, g)}{e(g^{(\alpha\xi+\gamma)y_jr}, g)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{(\alpha\xi+\beta+\gamma)x_jr}, g)}{e(g^{(\beta+\gamma)x_jr}, g)} \right)^{\frac{1}{x_j}} \right\}} \\
 &= m \cdot e(g_2 g_4, g)^r / (e(g, g)^{\beta r} \cdot e(g, g)^{\alpha \xi r}) = m
 \end{aligned}$$

[Twenty-eighth modification]

In the thirteenth embodiment, an example in which the public parameter includes five elements g, g_1, g_2, g_3, g_4 of system fixed values has been described, but the embodiment is not limited to such an example and the elements g_2, g_3, g_4 may not be generated and included in the public parameter. In this case, $g_2, g_3,$ and g_4 may be replaced by g in the thirteenth embodiment by setting $g_2=g, g_3=g,$ and $g_4=g$. This also applies to the embodiments and modifications described later.

[Twenty-ninth modification]

In the thirteenth embodiment, an example in which the

public parameter includes five elements g, g_1, g_2, g_3, g_4 of system fixed values has been described, but the embodiment is not limited to such an example and six or more elements of system fixed values may be included. If, for example,
5 six elements $g, g_1, g_2, g_3, g_4, g_5$ are included in the public parameter, g_2 may be replaced by g_2g_5 in the thirteenth embodiment by setting $g_2=g_2g_5$.

Also if, for example, seven elements $g, g_1, g_2, g_3, g_4, g_5, g_6$ are included in the public parameter, g_1 may be
10 replaced by g_1g_5 , g_2 may be replaced by g_2g_6 , and g_4 may be replaced by g_4g_5 in the thirteenth embodiment by setting $g_1=g_1g_5$, $g_2=g_2g_6$, and $g_4=g_4g_5$. This also applies to the embodiment and modifications described later.

(Fourteenth embodiment)

15 In the present embodiment, an example of updating a re-encryption key at certain intervals will be described. A time parameter L is used as a parameter representing a period. The time parameter is used in, of key setup processing, encryption processing, re-encryption key
20 generation processing, re-encryption processing, and decryption processing, the encryption processing, the re-encryption key generation processing, and the re-encryption processing. The time parameter is used such that if, for example, encryption processing is performed in a certain
25 period t_1 , the encryption processing described later is performed by setting $L=t_1$ and if encryption processing is performed in the next period t_2 , the encryption processing

described later is performed by setting $L=t_2$. Hereinafter,
key setup processing, encryption processing, re-encryption
key generation processing, re-encryption processing, and
decryption processing in the present embodiment will be
5 described. Portions common to the aforementioned
thirteenth embodiment may or may not be described by using
the same reference signs.

(1) Key setup processing

The public parameter is the same as that in the
10 thirteenth embodiment. If the identification information
of a re-encryption key generator 30 is i , a key generation
unit 14 generates a private key $x_i, y_i, z_i, w_i \in \mathbb{Z}_p^*$ of the
re-encryption key generator 30 and generates a public key
for encryption of the re-encryption key generator 30 using
15 the private key $sk_i = (x_i, y_i, z_i, w_i)$ (ST7).

$$pk_i = (X_i, Y_i, Y_{li}, Z_i, Z_{li}, W_i)$$

$$X_i = g^{x_i}, Y_i = g^{y_i}, Y_{li} = g^{y_i}, Z_i = g^{z_i}, Z_{li} = g^{z_i}, W_i = g^{w_i}$$

Also, the key generation unit 14 generates public
information for generating a re-encryption key of the re-
20 encryption key generator 30 using the private key sk_i
(ST7').

$$pi_i = (X_{3i}, Y_{2i})$$

$$X_{3i} = g_3^{x_i}, Y_{2i} = g_2^{y_i}$$

Similarly, if the identification information of a decryption device 50 is j , the key generation unit 14 generates a private key $sk_j = (x_j, y_j, z_j, w_j)$ of the decryption device 50 and generates a public key for encryption of the decryption device 50 using the private key sk_j (ST10).

$$pk_j = (X_j, Y_j, Y_{1j}, Z_j, Z_{1j}, W_j)$$

$$X_j = g^{x_j}, Y_j = g^{y_j}, Y_{1j} = g_1^{y_j}, Z_j = g^{z_j}, Z_{1j} = g_1^{z_j}, W_j = g^{w_j}$$

Also, the key generation unit 14 generates public information for generating a re-encryption key of the decryption device 50 using the private key sk_j (ST10').

$$pi_j = (X_{3j}, Y_{2j})$$

$$X_{3j} = g_3^{x_j}, Y_{2j} = g_2^{y_j}$$

(2) Encryption processing

Like in the thirteenth embodiment, an encryption parameter generation unit 23 of an encryption device 20 generates (ssk, svk) (ST21') and sets the verification key svk to the encrypted data C_1 ($C_1 = svk$).

The encryption parameter generation unit 23 also generates and sends out a random number $r \in \mathbb{Z}_p^*$ to a

ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data C_{2X} , C_{2Y} , C_{2Z} , C_{2Z1} , C_{2F} , C_3 , C_4 below from a message $m \in G_T$ as plaintext data using the random number r , the public key for encryption pk_i of the re-encryption key generator 30, and the time parameter L (ST22).

$$C_{2X} = X_i^r, C_{2Y} = Y_{li}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{li}^r, \\ C_{2F} = F_i(L)^r, C_3 = e(g_2 g_4, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$$

The function $F_i(L)$ is defined as shown below.

$$F_i(L) = g^L \cdot W_i = g^{L+w_i} \quad (L \in \mathbb{Z})$$

After step ST22 is completed, the ciphertext generation unit 24 generates a one-time signature σ as shown in the following formula for the time parameter L and the encrypted data C_3 , C_4 based on the signature generation function $\hat{S}$ in the public parameter and the signature key ssk generated in step ST21 (ST23).

$$\sigma = \hat{S}(ssk, (L, C_3, C_4))$$

Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_i = (L, C_1, C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3, C_4, \sigma)$ including the time parameter L , all the above encrypted data C_1 to C_4 , and the one-time signature σ and writes the obtained ciphertext data into a temporary data storage unit 21.

(3) Re-encryption key generation processing

A random number generation unit 36 generates and sends out three random numbers $\gamma, \delta_x, \delta_y \in \mathbb{Z}_p^*$ to a re-encryption key generation unit 34.

5 The re-encryption key generation unit 34 generates a re-encryption key R_{ijL} as shown in the following formulas using the random numbers $\gamma, \delta_x, \delta_y$, the private key sk_i of the re-encryption key generator 30 in a private key storage unit 31, the public information for generating a re-
10 encryption key pi_j of the decryption device 50 in a temporary data storage unit 32, and the time parameter L (ST32).

$$R_{ijL} = (R_{ijL1}, R_{ijL2}, R_{ijL3}, R_{ijL4}, R_{ijL5})$$

$$R_{ijL1} = (X_{3j} \cdot g^\gamma)^{\frac{1}{x_i}} \cdot F_i(L)^{\delta_x} = g^{\frac{\xi x_j + \gamma}{x_i} + (L + w_i) \delta_x}$$

$$R_{ijL2} = (Y_{2j} \cdot g^\gamma)^{\frac{1}{y_i}} \cdot F_i(L)^{\delta_y} = g^{\frac{\beta y_j + \gamma}{y_i} + (L + w_i) \delta_y}$$

$$R_{ijL3} = (X_{3j} \cdot Y_{2j} \cdot g^\gamma)^{\frac{1}{z_i}} = g^{\frac{\xi x_j + \beta y_j + \gamma}{z_i}}$$

$$R_{ijL4} = (g^{x_i})^{\delta_x} = g^{x_i \delta_x}$$

$$R_{ijL5} = (g^{y_i})^{\delta_y} = g^{y_i \delta_y}$$

15

In the above formula, g_1 is rewritten and denoted by

g^α , g_2 is rewritten and denoted by g^β , g_3 is rewritten and denoted by g^ξ , and g_4 is rewritten and denoted by $g^{\alpha\xi}$.

(4) Re-encryption processing

5 A re-encryption processing unit 44 verifies the ciphertext data C_i in a temporary data storage unit 42 using the public parameter, the time parameter L , and the following verification formula (ST41).

$$\begin{aligned} e(C_{2X}, u^{C_1} \cdot v) &= e(X_i, C_4), \\ e(C_{2Y}, u^{C_1} \cdot v) &= e(Y_{1i}, C_4), \\ e(C_{2Z}, u^{C_1} \cdot v) &= e(Z_i, C_4), \\ e(C_{2Z1}, u^{C_1} \cdot v) &= e(Z_{1i}, C_4), \\ e(C_{2F}, u^{C_1} \cdot v) &= e(F_i(L), C_4), \\ \hat{V}(C_1, \sigma, (L, C_3, C_4)) &= 1. \end{aligned}$$

10 The verification is successful if the six verification formulas shown above all hold and fails if any one of the six formulas does not hold.

If the verification is successful, a re-encryption parameter generation unit 45 generates and sends out four random numbers $s, t, k, h \in \mathbb{Z}_p^*$ to the re-encryption processing unit 44.

15 The re-encryption processing unit 44 generates, as shown in the following formulas, re-encrypted data C_{2X}' , C_{2X}'' , C_{2Y}' , C_{2Y}'' , C_{2Z}' , C_{2Z}'' , C_{2Z1}' , C_{2Z1}'' , C_{2F}' , C_{2F}'' , C_{5X} , C_{5Y} , C_{5Z} , C_{5FX} , C_{5FY} using these random numbers s, t, k, h , the

ciphertext data C_i in the temporary data storage unit 42, the re-encryption key R_{ijL} in the temporary data storage unit 42, and the time parameter L (ST42).

$$\begin{aligned}
 C'_{2X} &= X_i^s, C''_{2X} = C_{2X}^s = X_i^{rs}, C'_{2Y} = Y_{1i}^t, C''_{2Y} = C_{2Y}^t = Y_{1i}^{rt}, \\
 C'_{2Z} &= Z_i^k, C''_{2Z} = C_{2Z}^k = Z_i^{rk}, C'_{2Z1} = Z_{1i}^k, C''_{2Z1} = C_{2Z1}^k = Z_{1i}^{rk}, \\
 C'_{2F} &= F_i(L)^h, C''_{2F} = C_{2F}^h = F_i(L)^{rh}, \\
 C_{5X} &= R_{ijL1}^{\frac{1}{s}}, C_{5Y} = R_{ijL2}^{\frac{1}{t}}, C_{5Z} = R_{ijL3}^{\frac{1}{k}}, \\
 C_{5FX} &= R_{ijL4}^{\frac{1}{h}}, C_{5FY} = R_{ijL5}^{\frac{1}{h}}
 \end{aligned}$$

After step ST42 is completed, the re-encryption processing unit 44 generates re-encrypted data

$C_j = (C_1, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, C_3, C_4, \sigma)$ by replacing the encrypted data $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}$ in the ciphertext data C_i by all the above encrypted data/re-encrypted data C_{2X}' to C_{5FY} and writes the obtained ciphertext data into the temporary data storage unit 42.

(5) Decryption processing

A decryption processing unit 54 verifies the re-encrypted data C_j in a temporary data storage unit 52 using the public parameter, the public key for encryption pk_j of the local device 50, and the following verification formulas (ST51).

$$\begin{aligned}
e(C''_{2X}, u^{C_1} \cdot v) &= e(C'_{2X}, C_4) \\
e(C''_{2Y}, u^{C_1} \cdot v) &= e(C'_{2Y}, C_4) \\
e(C''_{2Z}, u^{C_1} \cdot v) &= e(C'_{2Z}, C_4) \\
e(C''_{2Z1}, u^{C_1} \cdot v) &= e(C'_{2Z1}, C_4) \\
e(C''_{2F}, u^{C_1} \cdot v) &= e(C'_{2F}, C_4) \\
e(C_{5Z}, C'_{2Z}) \cdot e(C_{5FX}, C'_{2F}) &= e(C_{5X}, C'_{2X}) \cdot e(Y_j, g_2) \\
e(C_{5Z}, C'_{2Z1}) \cdot e(C_{5FY}, C'_{2F}) &= e(C_{5Y}, C'_{2Y}) \cdot e(X_j, g_4) \\
\hat{V}(C_1, \sigma, (C_3, C_4)) &= 1
\end{aligned}$$

The verification is successful if the eight verification formulas shown above all hold and fails if any one of the eight formulas does not hold.

5 If the verification is successful, the decryption processing unit 54 decrypts the re-encrypted data C_j as shown in the following formula using the private key sk_j of the local device to obtain the message m (ST52).

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C''_{2Z}) \cdot e(C_{5FX}, C''_{2F})}{e(C_{5X}, C''_{2X})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C''_{2Z1}) \cdot e(C_{5FY}, C''_{2F})}{e(C_{5Y}, C''_{2Y})} \right)^{\frac{1}{x_j}} \right\}$$

10 That the decryption result of the above formula produces m can be checked by the following calculation. In the following calculation, g_1 is rewritten and denoted by g^α , g_2 is rewritten and denoted by g^β , g_3 is rewritten and denoted by g^ξ , and g_4 is rewritten and denoted by $g^{\alpha\xi}$.

$$\begin{aligned}
 & \frac{C_3}{\left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{C_3}{\left\{ \left(\frac{e(g^{\frac{\xi_j + \beta y_j + \gamma}{z_1 k}}, g^{z_1 r k}) \cdot e(g^{\frac{x_i \delta_x}{h}}, g^{(L+w_i)rh})}{e(g^{\left(\frac{\xi_j + \gamma}{x_i} + (L+w_i)\delta_x\right)^{\frac{1}{s}}}, g^{x_i r s})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\xi_j + \beta y_j + \gamma}{z_1 k}}, g^{a z_1 r k}) \cdot e(g^{\frac{\alpha y_i \delta_y}{h}}, g^{(L+w_i)rh})}{e(g^{\left(\frac{\beta y_j + \gamma}{y_i} + (L+w_i)\delta_y\right)^{\frac{1}{t}}}, g^{\alpha y_i r t})} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{C_3}{\left\{ \left(\frac{e(g^{(\xi_j + \beta y_j + \gamma)r}, g) \cdot e(g^{x_i \delta_x (L+w_i)r}, g)}{e(g^{(\xi_j + \gamma)r + (L+w_i)\delta_x x_i r}, g)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{(\xi_j + \beta y_j + \gamma)ar}, g) \cdot e(g^{\alpha y_i \delta_y (L+w_i)r}, g)}{e(g^{(\beta y_j + \gamma)ar + (L+w_i)\delta_y y_i ar}, g)} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{m \cdot e(g_2 g_4, g)^r}{(e(g, g)^{\beta r} \cdot e(g, g)^{\alpha \xi r})} = m
 \end{aligned}$$

In the present embodiment, the order of processing may be changed when appropriate. For example, the order of the decryption processing and the ciphertext verification processing may be changed. Similarly, the order of the re-encryption key generation processing may be changed to before the encryption processing.

According to the present embodiment, as described above, due to the configuration of generating the re-encryption key R_{ijL} based on the random numbers $\gamma, \delta_x, \delta_y$ and the time parameter L , in addition to the effects of the thirteenth embodiment, whether to transfer the decryption right can be decided in each period so that finer access control can be exercised. Accordingly, even if the decryption right of ciphertext for the user A is once transferred to the user B in a certain period, it is

possible not to grant the decryption right of the
ciphertext for the user A to the user B, that is, the
decryption right of the user B (to the ciphertext for the
user A) can be invalidated in the next period and therefore,
5 a file sharing system whose convenience is further improved
can be provided.

[Thirtieth modification]

In the fourteenth embodiment, an example in which
ciphertext data is generated by the encryption device 20,
10 the ciphertext data is re-encrypted by the re-encryption
device 40 to generate re-encrypted data, and the re-
encrypted data is decrypted by the decryption device 50 is
cited, but the eleventh embodiment may be changed to a form
in which the ciphertext data is decrypted without being re-
15 encrypted. In this case, only the key setup processing,
encryption processing, and decryption processing need to be
performed. The key setup processing in the present
modification is the same as the key setup processing in the
fourteenth embodiment. Hereinafter, the encryption
20 processing and the decryption processing in the present
modification will be described.

The difference between the encryption processing in
the present modification and the encryption processing in
the fourteenth embodiment is only in the last step. To
25 provide a description by making use of the above symbols,
the identification information of the decryption device 50
is assumed to be i for the sake of convenience. In this

case, the communication unit 22 of the encryption device 20 transmits the ciphertext data C_i in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25 (ST24').

5 The decryption device 50 verifies the ciphertext data C_i generated by the encryption device 20 in the same manner as in step ST41. If the verification is successful, the decryption device 50 decrypts the ciphertext data C_i as shown in the following formula using the private key sk_i to
10 obtain the message m .

$$m = \frac{C_3}{e(g_2 g_4, C_{2X})^{\frac{1}{x_i}}}$$

That the decryption result of the above formula produces m can be checked by the following calculation.

$$\begin{aligned} & m \cdot e(g_2 g_4, g)^r \bigg/ e(g_2 g_4, C_{2X})^{\frac{1}{x_i}} \\ = & m \cdot e(g_2 g_4, g)^r \bigg/ e(g_2 g_4, g^{x_i r})^{\frac{1}{x_i}} \\ = & m \cdot e(g_2 g_4, g)^r \bigg/ e(g_2 g_4, g)^r = m \end{aligned}$$

[Thirty-first modification]

In addition to the thirtieth modification, as will be described below, the fourteenth embodiment may also be changed to a form in which the ciphertext data is decrypted without being re-encrypted. Also in this case, only the key setup processing, encryption processing, and decryption processing need to be performed. The key setup processing and the decryption processing in the present modification are the same as the key setup processing and the decryption processing in the fourteenth embodiment respectively. Hereinafter, the encryption processing and the decryption processing in the present modification will be described. In the present modification, the identification information of the decryption device 50 is assumed to be j.

The encryption parameter generation unit 23 of the encryption device 20 generates (ssk, svk) in the same

manner as in step ST21 (ST21') and sets the verification key svk to the encrypted data C_1 ($C_1=svk$).

The encryption parameter generation unit 23 also generates and sends out eight random numbers

5 $r, s, t, k, h, \gamma, \delta_x, \delta_y \in \mathbb{Z}_p^*$ to the ciphertext generation unit 24.

The ciphertext generation unit 24 generates encrypted data $C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}''$, $C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, C_3, C_4$ below from a message $m \in G_T$ as plaintext data using the random numbers

10 $r, s, t, k, h, \gamma, \delta_x, \delta_y$, the public key pk_j of the decryption device 50, and the time parameter L (ST22').

$$\begin{aligned}
 C'_{2X} &= Y_j^s, C''_{2X} = Y_j^{rs}, C'_{2Y} = X_j^t, C''_{2Y} = X_j^{rt}, \\
 C'_{2Z} &= Y_j^k, C''_{2Z} = Y_j^{rk}, C'_{2Z1} = X_j^k, C''_{2Z1} = X_j^{rk}, \\
 C'_{2F} &= F_j(L)^h, C''_{2F} = F_j(L)^{rh}, \\
 C_3 &= e(g_2 g_4, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r, \\
 C_{5X} &= (g_4 \cdot g^\gamma \cdot F_j(L)^{\delta_y})^{\frac{1}{s}} = g^{\frac{\alpha\xi + \gamma + (L+w_j)\delta_y}{s}}, \\
 C_{5Y} &= (g_2 \cdot g^\gamma \cdot F_j(L)^{\delta_x})^{\frac{1}{t}} = g^{\frac{\beta + \gamma + (L+w_j)\delta_x}{t}}, \\
 C_{5Z} &= (g_4 \cdot g_2 \cdot g^\gamma)^{\frac{1}{k}} = g^{\frac{\alpha\xi + \beta + \gamma}{k}}, \\
 C_{5FX} &= Y_j^{\frac{\delta_y}{h}}, C_{5FY} = X_j^{\frac{\delta_x}{h}}
 \end{aligned}$$

After step ST22' is completed, the ciphertext generation unit 24 generates a one-time signature σ in the same manner as in step ST23.

5 Thereafter, the ciphertext generation unit 24 generates ciphertext data $C_j = (L, C_{2X}', C_{2X}'', C_{2Y}', C_{2Y}'', C_{2Z}', C_{2Z}'', C_{2Z1}', C_{2Z1}'', C_{2F}', C_{2F}'', C_{5X}, C_{5Y}, C_{5Z}, C_{5FX}, C_{5FY}, C_3, C_4, \sigma)$ including the time parameter L , all the above encrypted data C_1 to C_4 , and the one-time signature σ and writes the obtained ciphertext data into
10 the temporary data storage unit 21.

The communication unit 22 transmits the ciphertext data C_j in the temporary data storage unit 21 to the decryption device 50 under the control of the control unit 25.

15 The decryption device 50 verifies the ciphertext data C_j generated by the encryption device 20 in the same manner as in step ST51. If the verification is successful, the decryption device 50 decrypts the ciphertext data C_j as shown in the following formula using the private key sk_j to
20 obtain the message m .

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

That the decryption result of the above formula produces m can be checked by the following calculation. In the following calculation, g_1 is rewritten and denoted by
25 g^α , g_2 is rewritten and denoted by g^β , g_3 is rewritten and denoted by g^ξ , and g_4 is rewritten and denoted by $g^{\alpha\xi}$.

$$\begin{aligned}
 & \frac{C_3}{\left\{ \left(\frac{e(C_{5Z}, C_{2Z}^{\#}) \cdot e(C_{5FX}, C_{2F}^{\#})}{e(C_{5X}, C_{2X}^{\#})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}^{\#}) \cdot e(C_{5FY}, C_{2F}^{\#})}{e(C_{5Y}, C_{2Y}^{\#})} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{C_3}{\left\{ \left(\frac{e(g^{\frac{\alpha\xi+\beta+\gamma}{k}}, g^{y_jrk}) \cdot e(g^{\frac{y_j\delta_y}{h}}, g^{(L+w_j)rh})}{e(g^{\frac{\alpha\xi+\gamma+(L+w_j)\delta_y}{s}}, g^{y_jrs})} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{\frac{\alpha\xi+\beta+\gamma}{k}}, g^{x_jrk}) \cdot e(g^{\frac{x_j\delta_x}{h}}, g^{(L+w_j)rh})}{e(g^{\frac{\beta+\gamma+(L+w_j)\delta_x}{t}}, g^{x_jrl})} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{C_3}{\left\{ \left(\frac{e(g^{(\alpha\xi+\beta+\gamma)ry_j}, g) \cdot e(g^{y_j\delta_y(L+w_j)r}, g)}{e(g^{(\alpha\xi+\gamma)y_jr+(L+w_j)\delta_y y_jr}, g)} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(g^{(\alpha\xi+\beta+\gamma)x_jr}, g) \cdot e(g^{x_j\delta_x(L+w_j)r}, g)}{e(g^{(\beta+\gamma)x_jr+(L+w_j)\delta_x x_jr}, g)} \right)^{\frac{1}{x_j}} \right\}} \\
 &= \frac{m \cdot e(g_2 g_4, g)^r}{(e(g, g)^{\beta} \cdot e(g, g)^{\alpha\xi})} = m
 \end{aligned}$$

According to at least one of the thirteenth and
fourteenth embodiments described above, due to the
configuration of generating a re-encryption key based on a
random number, even if a server and a user collude with
each other, the decryption right can be prevented from
being re-transferred without permission of the transfer
source.

Also, due to the configuration of generating a re-
encryption key using public information for generating a
re-encryption key of the original transfer destination, no
re-encryption key can be generated without the public
information for generating a re-encryption key being
provided by the transfer destination. Therefore, a re-
encryption key can be generated with the permission of the

original transfer destination.

The method described in the embodiment can also be stored in a storage medium such as a magnetic disk (floppyTM disk, hard disk, or the like), an optical disk (CD-ROM, DVD, or the like), a magneto-optical disk (MO), or
5 a semiconductor memory as a program which can be executed by a computer and distributed.

As the storage medium, any configuration which is a computer-readable storage medium in which a program can be stored may be used regardless of a storage format.
10

An OS (operating system) which operates on a computer on the basis of an instruction of a program installed from the storage medium in the computer, database management software, and MW (middleware) such as network software may
15 execute a part of the processes to realize the embodiment.

Furthermore, the storage medium according to the present invention includes not only a medium independent of a computer but also a storage medium in which a program transmitted through a LAN, the Internet, or the like is
20 downloaded and stored or temporarily stored.

The number of storage media is not limited to one. A case in which the process in the embodiment is executed from a plurality of media is included in the storage medium according to the present invention. Any medium
25 configuration may be used.

A computer according to the present invention is to execute the processes in the embodiments on the basis of

the program stored in a storage medium. The computer may have any configuration such as one apparatus constituted by a personal computer or a system in which a plurality of apparatuses are connected by a network.

5 A computer in each embodiment includes not only a personal computer but also an arithmetic processing apparatus, a microcomputer, or the like included in an information processing apparatus. The computer is a generic name of an apparatus and a device which can realize
10 the functions of the present invention by a program.

 While certain embodiments have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the inventions. Indeed, the novel embodiments described herein may be
15 embodied in a variety of other forms; furthermore, various omissions, substitutions and changes in the form of the embodiments described herein may be made without departing from the spirit of the inventions. The accompanying claims and their equivalents are intended to cover such forms or
20 modifications as would fall within the scope and spirit of the inventions.

C L A I M S

1. A re-encryption key generator (30) that generates a re-encryption key to obtain re-encrypted data that can be decrypted by a second private key of a second user device
5 after ciphertext data obtained by encrypting plaintext data by a first public key of a first user device is re-encrypted without decryption, the re-encryption key generator comprising:

a first storage means (31) that stores a first private
10 key corresponding to the first public key;

a second storage means (32) that stores a second re-encryption key generation key of the second user device that is different from a second public key corresponding to the second private key; and

15 a re-encryption key generation means (34) that generates the re-encryption key based on the first private key and the second re-encryption key generation key.

2. The re-encryption key generator according to claim 1, wherein

20 the first public key is generated based on the first private key and a plurality of system fixed values and

the second public key is generated based on the second private key and the plurality of system fixed values.

25 3. The re-encryption key generator according to claim 2, wherein

when the first private key is represented by $sk_1=(x_1,y_1,z_1)$, the second private key is represented by

$sk_j=(x_j,y_j,z_j)$, the plurality of system fixed values are represented by g, g_1 , and g_2 (where $g, g_1, g_2 \in G$ when bilinear map groups as groups of an order p for which a bilinear map $e:G \times G \rightarrow G_T$ exists are represented by G, G_T), and
5 the first public key is represented by pk_i , the first public key pk_i contains data X_i, Y_{1i}, Z_i, Z_{1i} (where $X_i = g^{X_i}, Y_{1i} = g_1^{Y_i}, Z_i = g^{Z_i}, Z_{1i} = g_1^{Z_i}$),

when the second public key is represented by pk_j , the
10 second public key pk_j contains data X_j, Y_{1j}, Z_j, Z_{1j} (where $X_j = g^{X_j}, Y_{1j} = g_1^{Y_j}, Z_j = g^{Z_j}, Z_{1j} = g_1^{Z_j}$),

when a random number related to the second re-encryption key generation key is represented by π , a value
15 based on the random number π and the system fixed value g is represented by g_3 , the value based on the random number π and the system fixed value g_1 is represented by g_4 , the value based on the random number π and the system fixed value g_2 is represented by g_5 (where $g_3 = g^\pi, g_4 = g_1^\pi$, and
20 $g_5 = g_2^\pi$), and the second re-encryption key generation key is represented by rk_j ,

$$rk_j = (X_{3j}, Y_{5j})$$

holds (where

$$X_{3j} = g^{X_j}, Y_{5j} = g_5^{Y_j}$$

25),

when the plaintext data is represented by m (where

$m \in G_T$), the random number related to the ciphertext data is represented by r , and the ciphertext data is represented by C_i (where the bilinear map $e: G \times G \rightarrow G_T$ is represented by $e(,)$), the ciphertext data C_i contains data C_{2X} , C_{2Y} , C_{2Z} , C_{2Z1} , C_3 (where

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_3 = e(g_4 g_5, g)^r \cdot m.$$

),

when the random number related to the re-encryption key is represented by θ , an exponent related to the system fixed values g , g_2 is represented by β (where $g_2 = g^\beta$), and the re-encryption key is represented by R_{ij} ,

$$R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$$

holds (where

$$R_{ij1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} = g^{\frac{\pi(x_j + \theta)}{x_i}},$$

$$R_{ij2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} = g^{\frac{\pi(\beta y_j + \theta)}{y_i}},$$

$$R_{ij3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j + \beta y_j + \theta)}{z_i}}$$

),

when the re-encrypted data is represented by C_j (where three random numbers are represented by s , t , k), the re-encrypted data C_j includes data $C_{2X''}$, $C_{2Y''}$, $C_{2Z''}$, $C_{2Z1''}$,

$C_{5X}, C_{5Y}, C_{5Z}, C_3$ (where

$$C_{2X}'' = C_{2X}^s = X_i^{rs},$$

$$C_{2Y}'' = C_{2Y}^t = Y_{1i}^{rt},$$

$$C_{2Z}'' = C_{2Z}^k = Z_i^{rk},$$

$$C_{2Z1}'' = C_{2Z1}^k = Z_{1i}^{rk},$$

5

$$C_{5X} = R_{ij1}^{\frac{1}{s}}, C_{5Y} = R_{ij2}^{\frac{1}{t}}, C_{5Z} = R_{ij3}^{\frac{1}{k}}$$

), and

the plaintext data m , the re-encrypted data C_j , and the second private key sk_j are related by

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

10

4. The re-encryption key generator according to claim 2, wherein

when the first private key is represented by

$sk_i = (x_i, y_i, z_i, w_i)$, the second private key is represented by

15

$sk_j = (x_j, y_j, z_j, w_j)$, the plurality of system fixed values are

represented by g, g_1 , and g_2 (where $g, g_1, g_2 \in G$ when

bilinear map groups as groups of an order p for which a

bilinear map $e: G \times G \rightarrow G_T$ exists are represented by G, G_T), and

the first public key is represented by pk_i , the first

20

public key pk_i contains data $X_i, Y_{1i}, Z_i, Z_{1i}, W_i$ (where

$$X_i = g^{x_i}, Y_{1i} = g_1^{y_i}, Z_i = g^{z_i},$$

$$Z_{1i} = g_1^{z_i}, W_i = g^{w_i}$$

),

when the second public key is represented by pk_j , the second public key pk_j contains data $X_j, Y_{1j}, Z_j, Z_{1j}, W_j$

5 (where

$$X_j = g^{x_j}, Y_{1j} = g_1^{y_j}, Z_j = g^{z_j},$$

$$Z_{1j} = g_1^{z_j}, W_j = g^{w_j}$$

),

when a random number related to the second re-encryption key generation key is represented by π , a value based on the random number π and the system fixed value g is represented by g_3 , the value based on the random number π and the system fixed value g_1 is represented by g_4 , the value based on the random number π and the system fixed value g_2 is represented by g_5 (where $g_3 = g^\pi$, $g_4 = g_1^\pi$, and
10 $g_5 = g_2^\pi$), and the second re-encryption key generation key is represented by rk_j ,

$$rk_j = (X_{3j}, Y_{5j})$$

holds (where

$$X_{3j} = g_3^{x_j}, Y_{5j} = g_5^{y_j}$$

20),

when the plaintext data is represented by m (where $m \in G_T$), the random number related to the ciphertext data is

represented by r , a multiplicative group for the order p is represented by $\mathbb{Z}_p^*$, a time parameter showing a period of updating the re-encryption key is represented by L , and the ciphertext data is represented by C_i (where the bilinear map $e:G \times G \rightarrow G_T$ is represented by $e(,)$), the ciphertext data

C_i contains $C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_{2F}, C_3$ (where

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_{2F} = F_i(L)^r, C_3 = e(g_4 g_5, g)^r \cdot m,$$

$$F_i(L) = g^L \cdot W_i = g^{L+w_i} \quad (L \in \mathbb{Z}_p^*)$$

10),

when three random numbers related to the re-encryption key are represented by $\theta, \delta_x, \delta_y$, an exponent related to the system fixed values g, g_1 is represented by α , the exponent related to the system fixed values g, g_2 is represented by β (where $g_1 = g^\alpha$ and $g_2 = g^\beta$), and the re-encryption key is represented by R_{ijL} ,

$$R_{ijL} = (R_{ijL1}, R_{ijL2}, R_{ijL3}, R_{ijL4}, R_{ijL5})$$

holds (where

$$R_{ijL1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} \cdot g^{(L+w_i)\delta_x} = g^{\frac{\pi(x_j+\theta)}{x_i} + (L+w_i)\delta_x},$$

$$R_{ijL2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} \cdot g^{(L+w_i)\delta_y} = g^{\frac{\pi(\beta y_j+\theta)}{y_i} + (L+w_i)\delta_y},$$

$$R_{ijL3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j + \beta y_j + \theta)}{z_i}},$$

$$R_{ijL4} = g^{x_i \delta_x}, R_{ijL5} = g_1^{y_i \delta_y} = g^{\alpha y_i \delta_y}$$

),

when the re-encrypted data is represented by C_j (where four random numbers are represented by s, t, k, h), the re-encrypted data C_j contains data C_{2X}'' , C_{2Y}'' , C_{2Z}'' , C_{2Z1}'' ,

C_{2F}'' , C_{5X} , C_{5Y} , C_{5Z} , C_{5FX} , C_{5FY} , C_3 (where

$$C_{2X}'' = C_{2X}^s = X_i^{rs},$$

$$C_{2Y}'' = C_{2Y}^t = Y_{1i}^{rt}, C_{2Z}'' = C_{2Z}^k = Z_i^{rk},$$

$$C_{2Z1}'' = C_{2Z1}^k = Z_{1i}^{rk},$$

$$C_{2F}'' = C_{2F}^h = F_i(L)^{rh},$$

$$C_{5X} = R_{ijL1}^{\frac{1}{s}}, C_{5Y} = R_{ijL2}^{\frac{1}{t}}, C_{5Z} = R_{ijL3}^{\frac{1}{k}},$$

$$C_{5FX} = R_{ijL4}^{\frac{1}{h}}, C_{5FY} = R_{ijL5}^{\frac{1}{h}}$$

), and

the plaintext data m , the re-encrypted data C_j , and the second private key sk_j are related by

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

5. A re-encryption device (40) to obtain re-encrypted

data that can be decrypted by a second private key of a
second user device after ciphertext data obtained by
encrypting plaintext data by a first public key of a first
user device is re-encrypted by a re-encryption key without
5 decryption, the re-encryption device comprising:

a storage means (41) that stores the re-encryption key
generated by a re-encryption key generator in advance and

a re-encryption means (44) that obtains the re-
encrypted data by re-encrypting the ciphertext data by the
10 re-encryption key in the storage means without decrypting
the ciphertext data, wherein

the re-encryption key is generated based on a second
re-encryption key generation key of the second user device
that is different from a first private key corresponding to
15 the first public key and a second public key corresponding
to the second private key.

6. An encryption device (20) that generates
ciphertext data for a re-encryption device to obtain re-
encrypted data that can be decrypted by a second private
20 key of a second user device after the ciphertext data
obtained by encrypting plaintext data by a first public key
of a first user device is re-encrypted by a re-encryption
key without decryption, the encryption device comprising:

a storage means (21) that stores the first public key
25 and

an encryption means (24) that obtains the
ciphertext data by encrypting the plaintext data using the

first public key in the storage means, wherein

the re-encryption key is generated based on a second
re-encryption key generation key of the second user device
that is different from a first private key corresponding to
5 the first public key and a second public key corresponding
to the second private key.

7. A decryption device (50) that decrypts re-
encrypted data after obtaining the re-encrypted data that
can be decrypted by a second private key of a second user
10 device after ciphertext data obtained by encrypting
plaintext data by a first public key of a first user device
is re-encrypted by a re-encryption device based on a re-
encryption key without decryption, the decryption device
comprising:

15 a storage means (51) that stores the second private
key and

a decryption means (54) that obtains the plaintext
data by decrypting the re-encrypted data based on the
second private key in the storage means, wherein

20 the re-encryption key is generated based on a second
re-encryption key generation key of the second user device
that is different from a first private key corresponding to
the first public key and a second public key corresponding
to the second private key.

25 8. A program executed by a processor of a re-
encryption key generator (30) that generates a re-
encryption key to obtain re-encrypted data that can be

decrypted by a second private key of a second user device
after ciphertext data obtained by encrypting plaintext data
by a first public key of a first user device is re-
encrypted without decryption and stored in a non-transitory
5 computer-readable storage medium (M3), the program
comprising:

a first program code causing the processor to perform
processing (33) to write a first private key corresponding
to the first public key into a first storage means (31) of
10 the re-encryption key generator;

a second program code causing the processor to perform
the processing (33) to write a second re-encryption key
generation key of the second user device that is different
from a second public key corresponding to the second
15 private key into a second storage means (32) of the re-
encryption key generator; and

a third program code causing the processor to perform
processing (34) to generate the re-encryption key based on
the first private key and the second re-encryption key
20 generation key.

9. The program according to claim 8, wherein
the first public key is generated based on the first
private key and a plurality of system fixed values and
the second public key is generated based on the second
25 private key and the plurality of system fixed values.

10. The program according to claim 9, wherein
when the first private key is represented by

$sk_i=(x_i,y_i,z_i)$, the second private key is represented by
 $sk_j=(x_j,y_j,z_j)$, the plurality of system fixed values are
represented by g, g_1 , and g_2 (where $g, g_1, g_2 \in G$ when
bilinear map groups as groups of an order p for which a

5 bilinear map $e:G \times G \rightarrow G_T$ exists are represented by G, G_T), and
the first public key is represented by pk_i , the first
public key pk_i contains data X_i, Y_{1i}, Z_i, Z_{1i} (where

$$X_i = g^{X_i}, Y_{1i} = g_1^{Y_i}, Z_i = g^{Z_i}, Z_{1i} = g_1^{Z_i} \\),$$

10 when the second public key is represented by pk_j , the
second public key pk_j contains data X_j, Y_{1j}, Z_j, Z_{1j} (where

$$X_j = g^{X_j}, Y_{1j} = g_1^{Y_j}, Z_j = g^{Z_j}, Z_{1j} = g_1^{Z_j} \\),$$

when a random number related to the second re-
15 encryption key generation key is represented by π , a value
based on the random number π and the system fixed value g
is represented by g_3 , the value based on the random number
 π and the system fixed value g_1 is represented by g_4 , the
value based on the random number π and the system fixed
20 value g_2 is represented by g_5 (where $g_3=g^\pi, g_4=g_1^\pi$, and
 $g_5=g_2^\pi$), and the second re-encryption key generation key is
represented by rk_j ,

$$rk_j = (X_{3j}, Y_{5j})$$

holds (where

$$X_{3j} = g_3^{X_j}, Y_{5j} = g_5^{Y_j}$$

),

when the plaintext data is represented by m (where $m \in G_T$), the random number related to the ciphertext data is represented by r , and the ciphertext data is represented by C_i (where the bilinear map $e: G \times G \rightarrow G_T$ is represented by $e(,)$), the ciphertext data C_i contains data C_{2X} , C_{2Y} , C_{2Z} , C_{2Z1} , C_3 (where

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_3 = e(g_4 g_5, g)^r \cdot m.$$

10),

when the random number related to the re-encryption key is represented by θ , an exponent related to the system fixed values g , g_2 is represented by β (where $g_2 = g^\beta$), and the re-encryption key is represented by R_{ij} ,

15 $R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$

holds (where

$$R_{ij1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} = g^{\frac{\pi(x_j + \theta)}{x_i}},$$

$$R_{ij2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} = g^{\frac{\pi(\beta y_j + \theta)}{y_i}},$$

$$R_{ij3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j + \beta y_j + \theta)}{z_i}}$$

20),

when the re-encrypted data is represented by C_j (where

three random numbers are represented by s, t, k), the re-encrypted data C_j includes data $C_{2X}''', C_{2Y}''', C_{2Z}''', C_{2Z1}'''$, $C_{5X}, C_{5Y}, C_{5Z}, C_3$ (where

$$\begin{aligned} C_{2X}'' &= C_{2X}^s = X_i^{rs}, \\ C_{2Y}'' &= C_{2Y}^t = Y_{1i}^{rt}, \\ C_{2Z}'' &= C_{2Z}^k = Z_i^{rk}, \\ C_{2Z1}'' &= C_{2Z1}^k = Z_{1i}^{rk}, \end{aligned}$$

$$C_{5X} = R_{ij1}^{\frac{1}{s}}, C_{5Y} = R_{ij2}^{\frac{1}{t}}, C_{5Z} = R_{ij3}^{\frac{1}{k}}$$

), and

the plaintext data m , the re-encrypted data C_j , and the second private key sk_j are related by

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

11. The program according to claim 9, wherein

when the first private key is represented by

$sk_i = (x_i, y_i, z_i, w_i)$, the second private key is represented by $sk_j = (x_j, y_j, z_j, w_j)$, the plurality of system fixed values is represented by g, g_1 , and g_2 (where $g, g_1, g_2 \in G$ when bilinear map groups as groups of an order p for which a bilinear map $e: G \times G \rightarrow G_T$ exists are represented by G, G_T), and the first public key is represented by pk_i , the first public key pk_i contains data $X_i, Y_{1i}, Z_i, Z_{1i}, W_i$ (where

$$X_i = g^{x_i}, \quad Y_{1i} = g_1^{y_i}, \quad Z_i = g^{z_i}, \\ Z_{1i} = g_1^{z_i}, \quad W_i = g^{w_i}$$

),

when the second public key is represented by pk_j , the second public key pk_j contains data $X_j, Y_{1j}, Z_j, Z_{1j}, W_j$

5 (where

$$X_j = g^{x_j}, \quad Y_{1j} = g_1^{y_j}, \quad Z_j = g^{z_j}, \\ Z_{1j} = g_1^{z_j}, \quad W_j = g^{w_j}$$

),

when a random number related to the second re-encryption key generation key is represented by π , a value based on the random number π and the system fixed value g is represented by g_3 , the value based on the random number π and the system fixed value g_1 is represented by g_4 , the value based on the random number π and the system fixed value g_2 is represented by g_5 (where $g_3=g^\pi$, $g_4=g_1^\pi$, and $g_5=g_2^\pi$), and the second re-encryption key generation key is represented by rk_j ,

10

15

$$rk_j = (X_{3j}, Y_{5j})$$

holds (where

$$X_{3j} = g_3^{x_j}, \quad Y_{5j} = g_5^{y_j}$$

20

),

when the plaintext data is represented by m (where $m \in G_T$), the random number related to the ciphertext data is

represented by r , a multiplicative group for the order p is represented by $\mathbb{Z}_p^*$, a time parameter showing a period of updating the re-encryption key is represented by L , and the ciphertext data is represented by C_i (where the bilinear map $e:G \times G \rightarrow G_T$ is represented by $e(,)$), the ciphertext data C_i contains C_{2X} , C_{2Y} , C_{2Z} , C_{2Z1} , C_{2F} , C_3 (where

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, \\ C_{2F} = F_i(L)^r, C_3 = e(g_4 g_5, g)^r \cdot m,$$

$$F_i(L) = g^L \cdot W_i = g^{L+w_i} \quad (L \in \mathbb{Z}_p^*)$$

10),

when three random numbers related to the re-encryption key are represented by θ , δ_x , δ_y , an exponent related to the system fixed values g , g_1 is represented by α , the exponent related to the system fixed values g , g_2 is represented by β (where $g_1=g^\alpha$ and $g_2=g^\beta$), and the re-encryption key is represented by R_{ijL} ,

$$R_{ijL} = (R_{ijL1}, R_{ijL2}, R_{ijL3}, R_{ijL4}, R_{ijL5})$$

holds (where

$$R_{ijL1} = (X_{3j} \cdot g_3^\theta)^{1/x_i} \cdot g^{(L+w_i)\delta_x} = g^{\frac{\pi(x_j+\theta)}{x_i} + (L+w_i)\delta_x},$$

$$R_{ijL2} = (Y_{5j} \cdot g_3^\theta)^{1/y_i} \cdot g^{(L+w_i)\delta_y} = g^{\frac{\pi(\beta y_j+\theta)}{y_i} + (L+w_i)\delta_y},$$

$$R_{ijL3} = (X_{3j} \cdot Y_{5j} \cdot g_3^\theta)^{1/z_i} = g^{\frac{\pi(x_j + \beta y_j + \theta)}{z_i}},$$

$$R_{ijL4} = g^{x_i \delta_x}, R_{ijL5} = g^{y_i \delta_y} = g^{\alpha y_i \delta_y}$$

),

when the re-encrypted data is represented by C_j (where four random numbers are represented by s, t, k, h), the re-encrypted data C_j contains data C_{2X}'' , C_{2Y}'' , C_{2Z}'' , C_{2Z1}'' ,

C_{2F}'' , C_{5X} , C_{5Y} , C_{5Z} , C_{5FX} , C_{5FY} , C_3 (where

$$C_{2X}'' = C_{2X}^s = X_i^{rs},$$

$$C_{2Y}'' = C_{2Y}^t = Y_{1i}^{rt}, C_{2Z}'' = C_{2Z}^k = Z_i^{rk},$$

$$C_{2Z1}'' = C_{2Z1}^k = Z_{1i}^{rk},$$

$$C_{2F}'' = C_{2F}^h = F_i(L)^{rh},$$

$$C_{5X} = R_{ijL1}^{\frac{1}{s}}, C_{5Y} = R_{ijL2}^{\frac{1}{t}}, C_{5Z} = R_{ijL3}^{\frac{1}{k}},$$

$$C_{5FX} = R_{ijL4}^{\frac{1}{h}}, C_{5FY} = R_{ijL5}^{\frac{1}{h}}$$

), and

the plaintext data m , the re-encrypted data C_j , and the second private key sk_j are related by

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'') \cdot e(C_{5FX}, C_{2F}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'') \cdot e(C_{5FY}, C_{2F}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

12. A program executed by a processor of a re-

encryption device (40) to obtain re-encrypted data that can
be decrypted by a second private key of a second user
device after ciphertext data obtained by encrypting
plaintext data by a first public key of a first user device
5 is re-encrypted by a re-encryption key without decryption
and stored in a non-transitory computer-readable storage
medium (M4), the program comprising:

a first program code causing the processor to perform
processing (43) to write the re-encryption key generated by
10 a re-encryption key generator in advance into a storage
means (41) of the re-encryption device; and

a second program code causing the processor to perform
processing (44) to obtain the re-encrypted data by re-
encrypting the ciphertext data by the re-encryption key in
15 the storage means without decrypting the ciphertext data,
wherein

the re-encryption key is generated based on a second
re-encryption key generation key of the second user device
that is different from a first private key corresponding to
20 the first public key and a second public key corresponding
to the second private key.

13. A program executed by a processor of an
encryption device (20) that generates ciphertext data for a
re-encryption device (40) to obtain re-encrypted data that
25 can be decrypted by a second private key of a second user
device after the ciphertext data obtained by encrypting
plaintext data by a first public key of a first user device

is re-encrypted by a re-encryption key without decryption and stored in a non-transitory computer-readable storage medium (M2), the program comprising:

5 a first program code causing the processor to perform processing (22) to write the first public key into a storage means (21) of the encryption device; and

 a second program code causing the processor to perform processing (24) to obtain the ciphertext data by encrypting the plaintext data using the first public key in the
10 storage means, wherein

 the re-encryption key is generated based on a second re-encryption key generation key of the second user device that is different from a first private key corresponding to the first public key and a second public key corresponding
15 to the second private key.

14. A program executed by a processor of a decryption device (50) that decrypts re-encrypted data after obtaining the re-encrypted data that can be decrypted by a second private key of a second user device after ciphertext data
20 obtained by encrypting plaintext data by a first public key of a first user device is re-encrypted by a re-encryption device (40) based on a re-encryption key without decryption and stored in a non-transitory computer-readable storage medium (M5), the decryption device comprising:

25 a first program code causing the processor to perform processing (53) to write the second private key into a storage means (51) of the decryption device; and

a second program code causing the processor to perform processing (54) to obtain the plaintext data by decrypting the re-encrypted data based on the second private key in the storage means, wherein

5 the re-encryption key is generated based on a second re-encryption key generation key of the second user device that is different from a first private key corresponding to the first public key and a second public key corresponding to the second private key.

10 15. A re-encryption key generator (30) that generates a re-encryption key to obtain re-encrypted data that can be decrypted by a second private key of a second user device after ciphertext data obtained by encrypting plaintext data by a public key for encryption of a first user device is
15 re-encrypted without decryption, the re-encryption key generator comprising:

a first storage means (31) that stores a first private key corresponding to the public key for encryption;

20 a second storage means (32) that stores public information for generating a re-encryption key corresponding to the second private key;

a first random number generation means (36) that generates a first random number; and

25 a re-encryption key generation means (34) that generates the re-encryption key based on the first private key, the public information for generating a re-encryption key, and the first random number.

16. The re-encryption key generator according to claim 15, wherein

the re-encryption key generation means generates the re-encryption key based on the first private key, the public information for generating a re-encryption key, and the first random number independently of ciphertext data before re-encryption.

17. The re-encryption key generator according to claim 15, wherein

the public key for encryption is generated based on the first private key and a plurality of system fixed values and

the public information for generating a re-encryption key is generated based on the second private key and the plurality of system fixed values.

18. The re-encryption key generator according to claim 16, wherein

when the first private key is represented by $sk_i=(x_i, y_i, z_i)$, the second private key is represented by $sk_j=(x_j, y_j, z_j)$, the first random number is represented by γ , the plurality of system fixed values are represented by g, g_1, g_2, g_3, g_4 (where $g, g_1, g_2, g_3, g_4 \in G$ when bilinear map groups as groups of an order p for which a bilinear map $e: G \times G \rightarrow G_T$ exists are represented by G, G_T), exponents related to the system fixed values are represented by α, β, ξ (where $g_1=g^\alpha, g_2=g^\beta, g_3=g^\xi$, and $g_4=g^{\alpha\xi}$), and the re-encryption key is represented by R_{ij} ,

$$R_{ij} = (R_{ij1}, R_{ij2}, R_{ij3})$$

holds (where

$$R_{ij1} = g^{\frac{\xi x_j + \gamma}{x_i}}, R_{ij2} = g^{\frac{\beta y_j + \gamma}{y_i}}, R_{ij3} = g^{\frac{\xi x_j + \beta y_j + \gamma}{z_i}}$$

),

5 when the public key for encryption of the first user device is represented by pk_i ,

$$pk_i = (X_i, Y_{1i}, Z_i, Z_{1i})$$

holds (where

$$X_i = g^{x_i}, Y_{1i} = g_1^{y_i}, Z_i = g^{z_i}, Z_{1i} = g_1^{z_i}$$

10),

when the public information for generating a re-encryption key corresponding to the first private key is represented by pi_i ,

$$pi_i = (X_{3i}, Y_{2i})$$

15 holds (where

$$X_{3i} = g_3^{x_i}, Y_{2i} = g_2^{y_i}$$

),

when the public key for encryption of the second user device is represented by pk_j ,

$$pk_j = (X_j, Y_{1j}, Z_j, Z_{1j})$$

holds (where

$$X_j = g^{x_j}, Y_{1j} = g_1^{y_j}, Z_j = g^{z_j}, Z_{1j} = g_1^{z_j}$$

),

5 when the public information for generating a re-
encryption key corresponding to the second private key is
represented by pi_j ,

$$pi_j = (X_{3j}, Y_{2j})$$

holds (where

$$X_{3j} = g^{x_j}, Y_{2j} = g^{y_j}$$

10

),

when the plaintext data is represented by m (where
 $m \in G_T$), a second random number is represented by r , and the
ciphertext data is represented by C_i (where the bilinear
15 map $e: G \times G \rightarrow G_T$ is represented by $e(g, h)$ (where $g, h \in G$)),

$$C_i = (C_{2X}, C_{2Y}, C_{2Z}, C_{2Z1}, C_3)$$

holds (where

$$C_{2X} = X_i^r, C_{2Y} = Y_{1i}^r, C_{2Z} = Z_i^r, C_{2Z1} = Z_{1i}^r, C_3 = e(g_2 g_4, g)^r \cdot m$$

),

20 when the re-encrypted data is represented by C_j (where
third to fifth random numbers are represented by s, t, k),

$$C_j = (C_{2X}''', C_{2Y}''', C_{2Z}''', C_{2Z1}''', C_{5X}, C_{5Y}, C_{5Z}, C_3)$$

holds (where

$$\begin{aligned} C_{2X}'' &= C_{2X}^s = X_i^{rs}, \\ C_{2Y}'' &= C_{2Y}^t = Y_{1i}^{rt}, \\ C_{2Z}'' &= C_{2Z}^k = Z_i^{rk}, \\ C_{2Z1}'' &= C_{2Z1}^k = Z_{1i}^{rk}, \\ C_{5X} &= R_{ij1}^{\frac{1}{s}}, \quad C_{5Y} = R_{ij2}^{\frac{1}{t}}, \quad C_{5Z} = R_{ij3}^{\frac{1}{k}} \end{aligned}$$

), and

the plaintext data m , the re-encrypted data C_j , and
the second private key sk_j are related by

$$m = C_3 / \left\{ \left(\frac{e(C_{5Z}, C_{2Z}'')}{e(C_{5X}, C_{2X}'')} \right)^{\frac{1}{y_j}} \cdot \left(\frac{e(C_{5Z}, C_{2Z1}'')}{e(C_{5Y}, C_{2Y}'')} \right)^{\frac{1}{x_j}} \right\}$$

19. A re-encryption device (40) capable of
communicating with a re-encryption key generator (30) that
generates a re-encryption key to obtain re-encrypted data
that can be decrypted by a second private key of a second
user device after ciphertext data obtained by encrypting
plaintext data by a public key for encryption of a first
user device is re-encrypted without decryption, the re-
encryption device comprising:

a storage means (41) that stores the re-encryption key
received from the re-encryption key generator;

a re-encryption means (44) that obtains the re-

encrypted data by re-encrypting the ciphertext received from the first user device without decryption by the re-encryption key in the storage means; and

5 a means (43) that transmits the obtained re-encrypted data to the second user device, wherein

the re-encryption key is generated based on a first private key corresponding to the public key for encryption, public information for generating a re-encryption key corresponding to the second private key, and a first random
10 number generated by the re-encryption key generator.

20. The re-encryption device according to claim 19, wherein

the re-encryption key is generated independently of the ciphertext data before re-encryption.

15 21. A program executed by a processor of a re-encryption key generator (30) that generates a re-encryption key to obtain re-encrypted data that can be decrypted by a second private key of a second user device after ciphertext data obtained by encrypting plaintext data
20 by a public key for encryption of a first user device is re-encrypted without decryption and stored in a non-transitory computer-readable storage medium (M3), the program comprising:

25 a first program code causing the processor to perform processing to write a first private key corresponding to the public key for encryption into a first storage means (31) of the re-encryption key generator;

a second program code causing the processor to perform the processing to write public information for generating a re-encryption key corresponding to the second private key into a second storage means (32) of the re-encryption key generator;

a third program code causing the processor to perform processing (36) to generate a first random number; and

a fourth program code causing the processor to perform processing (34) to generate the re-encryption key based on the first private key, the public information for generating a re-encryption key, and the first random number.

22. The program according to claim 21, wherein the processor is caused to perform the processing to generate the re-encryption key independently of the ciphertext data before re-encryption.

23. A program executed by a processor of a re-encryption device (40) capable of communicating with a re-encryption key generator (30) that generates a re-encryption key to obtain re-encrypted data that can be decrypted by a second private key of a second user device after ciphertext data obtained by encrypting plaintext data by a public key for encryption of a first user device is re-encrypted without decryption and stored in a non-transitory computer-readable storage medium, the program comprising:

a first program code causing the processor to perform processing to write the re-encryption key received from the

re-encryption key generator into a storage means (41) of the re-encryption device;

5 a second program code causing the processor to perform processing (44) to obtain the re-encrypted data by re-encrypting the ciphertext data received from the first user device by the re-encryption key in the storage means without decryption; and

10 a third program code causing the processor to perform processing (43) to transmit the obtained re-encrypted data to the second user device, wherein

15 the re-encryption key is generated based on a first private key corresponding to the public key for encryption, public information for generating a second re-encryption key corresponding to the second private key, and a first random number generated by the re-encryption key generator.

24. A program executed by a processor of an encryption device (20) capable of communicating with a re-encryption device (40) that uses a re-encryption key to obtain re-encrypted data that can be decrypted by a second private key of a second user device after ciphertext data obtained by encrypting plaintext data by a public key for encryption of a first user device is re-encrypted without decryption and stored in a non-transitory computer-readable storage medium (M2), the program comprising:

25 a first program code causing the processor to perform processing (22) to write the public key for encryption into a storage means (21) of the encryption device;

a second program code causing the processor to perform processing (24) to obtain the ciphertext data by encrypting the plaintext data using the public key for encryption in the storage means; and

5 a third program code causing the processor to perform processing (22) to transmit the obtained ciphertext data to the re-encryption device, wherein

the re-encryption key is generated based on a first private key corresponding to the public key for encryption,
10 public information for generating a re-encryption key corresponding to the second private key, and a first random number generated by a re-encryption key generator that generates the re-encryption key.

25. A program executed by a processor of a decryption
15 device (50) that decrypts re-encrypted data obtained from the re-encryption device after obtaining the re-encrypted data that can be decrypted by a second private key of a second user device after ciphertext data obtained by encrypting plaintext data by a public key for encryption of
20 a first user device is re-encrypted by a re-encryption device (40) based on a re-encryption key without decryption and stored in a non-transitory computer-readable storage medium (M5), the program comprising:

a first program code causing the processor to perform
25 processing (53) to write the second private key into a storage means (51) of the decryption device; and

a second program code causing the processor to perform

processing (54) to obtain the plaintext data by decrypting
the re-encrypted data based on the second private key in
the storage means after receiving the re-encrypted data
from the re-encryption device, wherein

5 the re-encryption key is generated based on a first
private key corresponding to the public key for encryption,
public information for generating a re-encryption key
corresponding to the second private key, and a first random
number generated by a re-encryption key generator that
10 generates the re-encryption key.

26. The program according to any one of claims 23 to
25, wherein

the re-encryption key is generated independently of
the ciphertext data before re-encryption.

15