DESCRIPTION

METHOD AND DEVICE FOR CALCULATING A FUNCTION FROM A LARGE NUMBER OF INPUTS

Technical Field

The present invention relates to a method of calculating, when inputs for a given function are dispersed and held in a plurality of devices, an output of this function while these devices are working together, and more particularly to a method and system for performing a calculation by a fixed number of times irrespective of a function in which the number of times for each device to perform communication with another device is given.

Background Art

As a prior art regarding a method of calculating, when inputs for a given function are dispersed and held in a plurality of devices, an output of this function while these devices are working together, a method is proposed by Beaver, Micali, and Rogaway in paper "D. Beaver, S. Micali, and P. Rogaway, 'The round complexity of secure protocols', Annual ACM Symposium on Theory of Computing 22, pages 503-513, 1990". This paper is hereinafter referred to as Non-patent Document 1.

The technology disclosed in Non-patent Document 1 relates to a method of calculating, when the number of calculators u_α who are connected to one another via a network is λ, each of the calculators has a secret input x_α, and an arbitrary function g is given, an output $g(x_1, ..., \lambda)$ while the calculators work together, wherein the secret of each of the calculators is not leaked beyond $g(x_1, ..., x_\lambda)$ and the number of times for performing communication necessary for

the calculation with is a fixed number. The technology disclosed in Non-patent Document 1 will be described with reference to Figs. 1, 2 and 19.

[Garbled circuit]

[Syntax]

A circuit f includes m logic gates. Each gate is denoted by symbols $G_1,...,G_i,...,G_m$. As shown in Fig. 19, each gate has two inputs and one output. Each output may be input to a plurality of gates. An output line of G_k is generally input to a plurality of gates but all signals flowing through the line have the same value of 0 or 1. Then, all line output from the gate G_k are referred to as w_k. The number of line input to the circuit f is n, which is expressed as follows.

{wk} k = m+1, ..., m+n. Then, $w_1,...,w_l$ denote the output of the circuit f.

The number of calculators is λ, and a set of the calculators is expressed as follows.

$$\left\{u^{(\alpha)}\right\}_{\alpha=1,\ ...,\ \lambda}$$

The number of bits input by $u^{(\alpha)}$ to the circuit f is I_α.

With respect to $(\Sigma_{\alpha=1}^{\lambda} I_\alpha = \lambda)$. $k = m+1,\ ...,\ m+n$ where the sum of those bits is set as n, a bit input to each w_k is represented by b_k, and each of the bits is allocated to $u^{(\alpha)}$ by the number of I_α in the following manner. That is, $u^{(\alpha)}$ determines the following set.

$$\{b_K \in \{0,1\}\}k = m + \Sigma_{\beta=1}^{\alpha-1} I_\beta + 1,\ ...,\ m + \Sigma_{\beta=1}^{\alpha} I_\beta$$

When outputs of the gates G_i and G_j are input to the gate G_k, a relation between the output b_j of G_j and the output b_k of G_k is represented as follows.

$$b_K = b_i \circledcirc_{G[k]} b_j$$

Then, ⊡ denotes exclusive OR of the bit and · denotes a character string sequences.

t denotes a safety variable, and G, H, and F denote pseudorandom number generating devices for output character strings of tλ bits.

[Construction]

A protocol is roughly divided into three processes which are (1) an input process 402, (2) a parallel construction process 400 of a garbled circuit based on a calculation performed by a large number of people, and (3) a result output process 401 for performing an input disclosure and a circuit calculation.

The input process 402 is performed in the following manner. Information on a circuit for performing the calculation, information on another calculator, and input data of each device are input to each device.

The parallel construction process 400 of the garbled circuit is performed in the following manner. In a procedure of this process, as shown in Fig. 2, a phase 502 in which λ computers 501 individually performs the calculation and a phase 503 in which all the computers perform communication with one another are alternately performed. Then, the number of performing the repetition is set as a fixed number 504, and whatever function is desired to be calculated, the following process can be completed. Also, in each communication phase, each of the computers transmits data to all of the other computers. In order to generate the data transmitted at this time, transmission data of other computers in the same communication phase of this transmission should not be needed. That is, when there is a transmission which should wait for data of other computers, the communication phase in which this transmission is performed is counted as a different communication phase from the communication phase in which data is waited for.

[1] In cooperation with one another, the calculators uniquely and randomly generate a set of character strings of bits and a set of bits so that these sets are secretly dispersed to all the calculators.

$\{s^{\alpha}{}_{k'}s^{\alpha}{}_{k} \in_R \{0,1\}^t\}$ k=1, ..., m+n; α=1, ..., λ

$\{\rho_k \in_R \{0,1\}\}$

Wherein

$S_k := s_k{}^1 \cdot s_k{}^2 \cdot \ldots s_k{}^{\lambda}$

$S'_k := s'_k{}^1 \cdot s'_k{}^2 \cdot \ldots s'_k{}^{\lambda}$.

Regarding $\{S_k\},\{\rho_k\}$, if $\lambda_k \square b_k=0$, in the calculation phase of the circuit, S_k is made public, and if $\lambda_k \square b_k=1$, S'_k is made public.

[2] For each of the calculators u_α, the following data is revealed.

$\{s^{\alpha}{}_{k}\}$ k=1, ..., m+n

[3] With respect to k=1, ..., m+n, each of the calculators u_α calculates the following character strings of tλ bits.

$g^{\alpha}{}_{k} = G(s^{\alpha}{}_{k})$
$g'^{\alpha}{}_{k} = G(s'^{\alpha}{}_{k})$
$h^{\alpha}{}_{k} = H(s^{\alpha}{}_{k})$
$h'^{\alpha}{}_{k} = H(s'^{\alpha}{}_{k})$
$f^{\alpha}{}_{k} = F(s^{\alpha}{}_{k})$
$f'^{\alpha}{}_{k} = F(s'^{\alpha}{}_{k})$

Then, each of the calculators u_α commits the following data to prove to the other calculators that these values are calculated correctly.

$\{g^{\alpha}{}_{k'}, g'^{\alpha}{}_{k'}, h^{\alpha}{}_{k'}, h'^{\alpha}{}_{k'}, f^{\alpha}{}_{k'}, f'^{\alpha}{}_{k'}\}_k$

[4] With respect to k=1, ..., m+n, the calculators secretly perform the following calculation in a dispersed manner.

$\sigma_k{}^1 * \ldots \sigma_k{}^{\lambda} = S_k$ if $\lambda_k \square b_k = 0$
$\sigma_k{}^1 * \ldots \sigma_k{}^{\lambda} = S'_k$ if $\lambda_k \square b_k = 1$

[5] In cooperation with one another, with respect to k=1, ..., m+n, the calculators secretly perform the following calculation in a dispersed manner.

$A_k = g_i^1 \square \ldots \square g_i^\lambda \ \square g_j^1 \square \ldots \square g_j^\lambda \ \square S_k$ if $\rho_i \circledcirc_{G[k]} \rho_j = \rho_k$

$A_k = g_i^1 \square \ldots \square g_i^\lambda \ \square g_j^1 \square \ldots \square g_j^\lambda \ \square S'_k$ if $\rho_i \circledcirc_{G[k]} \rho_j \neq \rho_k$

$B_k = h_i^1 \square \ldots \square h_i^\lambda \ \square g'^1_j \square \ldots \square g'^\lambda_j \ \square S_k$ if $\rho_i \circledcirc_{G[k]} \rho_j = \rho_k$

$B_k = h_i^1 \square \ldots \square h_i^\lambda \ \square g'^1_j \square \ldots \square g'^\lambda_j \ \square S'_k$ if $\rho_i \circledcirc_{G[k]} \rho'_j \neq \rho_k$

$C_k = g'^1_i \square \ldots \square g'^\lambda_i \ \square h_j^1 \square \ldots \square h_j^\lambda \ \square S_k$ if $\rho'_i \circledcirc_{G[k]} \rho_j = \rho_k$

$C_k = g'^1_i \square \ldots \square g'^\lambda_i \ \square h_j^1 \square \ldots \square h_j^\lambda \ \square S'_k$ if $\rho'_i \circledcirc_{G[k]} \rho_j \neq \rho_k$

$D_k = h'^1_i \square \ldots \square h'^\lambda_i \ \square g'^1_j \square \ldots \square g'^\lambda_j \ \square S_k$ if $\rho'_i \circledcirc_{G[k]} \rho'_j = \rho_k$

$D_k = h'^1_i \square \ldots \square h'^\lambda_i \ \square g'^1_j \square \ldots \square g'^\lambda_j \ \square S'_k$ if $\rho'_i \circledcirc_{G[k]} \rho'_j \neq \rho_k$

It should be noted that a signal input to the gate G_k is outputs of the gate G_i and the gate G_j. This state is shown in Fig. 19. The disclosure of the input and the generation of the circuit in the result output process 401 are performed as follows.

[1] The calculators reveal the following data.

$\{\rho_k\}_{k=1,\ldots,l}$

$\{f_k^\alpha\}_{k=1,\ldots,m+n;\ \alpha=1,\ldots,\lambda}$

$\{\sigma_k^1 * \ldots * \sigma_k^\lambda\}_{k=1,\ldots,m+n}$

$\{A_k, B_k, C_k, D_k\}\}_{k=1,\ldots,m+n}$

[2] With respect to k=1, ..., m+n, in an order from k which is closer to the input of the circuit, from S_i or S'_i, and S_j or S'_j, S^*_k is obtained as follows. This refers to S_k or S'_k.

$S_k^* = A_k \square g_i^1 \square \ldots \square g_i^\lambda \ \square g_j^1 \square \ldots \square g_j^\lambda$ if S_i, S_j are processed

$S_k^* = B_k \square h_i^1 \square \ldots \square h_i^\lambda \ \square g'^1_j \square \ldots \square g'^\lambda_j$ if S_i, S'_j are processed

$S_k^* = C_k \square g'^1_i \square \ldots \square g'^\lambda_i \ \square h_j^1 \square \ldots \square h_j^\lambda$ if S'_i, S'_j are processed

$S_k^* = D_k \square h'^1_i \square \ldots \square h'^\lambda_i \ \square g'^1_j \square \ldots \square g'^\lambda_j$ if S'_i, S'_j are processed

[3] With respect to all of α=1, ..., λ; k=1, ..., m+n, by checking the following, $S^*_k=S_k$ or $S^*_k=S'_k$ is confirmed.

$f^\alpha_{k\theta}=F(s^\alpha_k)$,

$f'^\alpha_k=F(s'^\alpha_k)$

[4] With respect to k=1, ..., m+n, when all the calculators obtain S_k, $\rho_k + b_k = 0$ is established, and when all the calculators obtain S'_k, $\rho_k + b_k = 1$ is established, thereby finding out b_k.

As other prior art for such a method as described in the section of Technical Field, there is a method proposed by Ishai and Kushilevitz in paper "Y. Ishai and E. Kushilevitz, 'Randomizing Polynomials: A new Representation with Applications to Round-Efficient Secure Computation', IEEE Symposium on Foundations of Computer Science 2000, pages 294-304". Hereafter, this paper is referred to as Non-patent Document 2. The prior art of Non-patent Document 2 will be described with reference to Figs. 3 and 4.

[Randomizing polynomial]

Non-patent Document 2 proposes a method of expressing a given function by a low order polynomial on the finite field. In particular, Non-patent Document 2 demonstrates that an arbitrary function can be expressed by a third polynomial. Evaluation of a low order polynomial can be performed by performing a round by a fixed number of times. In general, the function can be expressed in various forms such as a circuit.

A blanching problem described next can be expressed by a general function. A blanching problem BP = (G,φ, s,t) is referred to a mod-p blanching problem. G = (V, E) is a directed graph. φ is a labeling function for labeling one of 1, x^1_i, and the negation x^0_i to each of the sides. Then, s and t are special apexes.

When an input $x = (x_1, ..., x_n)$ is given, from the labeling function φ, a partial graph Gx of G is given. A value of a Boolean function f calculated by is f(x) = 0 when a remainder obtained through division of the number of routes connecting s with t in G_x by p is 0, and otherwise the value is f(x) = 1. The magnitude of BP is set as the number of the apexes of G.

The magnitude of BP is set as l. When a l\timesl adjacency matrix of the

partial graph Gx is expressed as Hx, the number of routes connecting s-t is obtained as follows.

$$(I+H_x + H_x^2+\cdots)_{st} = ((1-H_x)^{-1})_{st} \bmod p$$

$$= \det M_x / \det (I-H_x) \} \bmod p$$

Wherein M_x is a matrix obtained by excluding a row s and a column t from the matrix (I-Hx). Therefore, the following data is found out.

$f(x) = 0 \Leftrightarrow \text{rank}\ (M_x) = I-1$

$f(x) = 0 \Leftrightarrow \text{rank}\ (M_x) = I$

Then, M_x includes an at most first order component with respect to x.

[Calculation method]

A method of obtaining f(x) by using the randomizing polynomial method when the Boolean function f is given and the input is distributed to a plurality of calculators.

As shown in Fig. 3,

[1] Information on a function to be calculated, information on another calculator, and input data of each device are input to each device (605).

[2] BP is constructed which corresponds to f (600).

[3] The following processes are performed in parallel by the sufficient number of times.

[Process]

As shown in Fig. 4,

All the calculators disperse each component to uniquely and randomly generate lxl matrices R_1, R_2 (603) for calculating $R_1M_xR_2$ which is a product of three matrices R_1, M_x, R_2 (604).

Each component is an at most third expression of components of R_1, R_2, x.

[4] From all the values of rank$R_1M_xR_2$, it is presumed whether or not rank of M_x is I. When the probability that rank of M_x is I is high, 1 is output, and otherwise 0 is output (602).

In the above-mentioned method, when rank (M_x) = rank (M'_x), distributions of $R_1M_xR_2$ and $R_1M'_xR_2$ become the same, a new matter is not leaked other than f(x) with respect to x.

Furthermore, when rank (M_x) = l with respect to any l, the probability of rank ($R_1M_xR_2$) = l is larger than 0.08. Thus, the number of times for performing the process of Item 2 does not rely on l.

[Calculation amount and communication amount]

In the method using the garbled circuit, the calculation with respect to each gate is individually performed, and the entire calculation amount and communication amount are proportional to the number of gates. t-n threshold dispersion (proportional to $2t^2$. The calculation in the t-n threshold dispersion refers to a calculation method in which the secret is dispersed to n calculators. Among the calculators, unless t calculators gather the data which each of them knows by itself, it is impossible to find out the dispersed secret or meaningful data in the middle of the calculation.

In the method using the randomizing polynomial, in the case where the t-n threshold dispersion is performed, the round number becomes 2(3) in proportion to t^2 and the square of the magnitude of BP.

The communication amount and calculation amount in the randomizing polynomial method are proportional to the at most first order of the number of gates. Moreover, a coefficient of the highest order is substantially lower than that of the randomizing polynomial method and therefore efficient.

However, here, particular attention is paid to the case where t > n/2 is satisfied in the t-n threshold dispersion and a third party demands the verification of the calculation validity. In such a case, it is obviously applicable to extend the above-mentioned method. The result of the extension shows that the entire communication amount and calculation amount in the method using the garbled circuit are proportional to the number of gates and t3. When method using the

randomizing polynomial is used, the communication amount and calculation amount are proportional to 1.5-th power of the number of gates. When the number of gates is larger, the method is not efficient.

A first problem resides in that the method of Non-patent Document 1 requires an enormous calculation amount of each calculator and an enormous calculation amount of a verifier who verifies the calculation validity.

This is because as each calculator needs to calculate the output of the pseudorandom number generating device, it is necessary to prove the calculation correctness while the calculation result is hidden.

A second problem resides in that the method of Non-patent Document 2 also requires an enormous calculation amount of each calculator and an enormous calculation amount of the verifier who verifies the calculation validity.

This is because the calculation amount performed by each calculator is in proportion to 1.5-th power of the number of gates in the case of expressing the function by the circuit and often the number of gates is extremely large. Thus the entire calculation amount becomes enormous.

Disclosure of Invention

An object of the present invention provides is to provide a method and system for performing a calculation by fixed times irrespective of a function in which the number of computers is suppressed to be proportional to the number of gates even when the number of gates is larger, the computers do not need to calculate outputs of pseudorandom number generating devices whose calculation validity should be proved, and the number of performing communication by the computer becomes a fixed number irrespective of a function.

According to an aspect of the present invention, there is provided a method of calculating a value of a given function by using an apparatus that

includes a plurality of computers, including:

an input process; and

an output process,

characterized in that the input process inputs a circuit and an input bit to the circuit to the plurality of computers, and

one of the computers firstly performs calculation and transmits the calculation result to another computer and the another computer which has received the calculation result performs the next calculation such that calculation is performed by one computer after another, and when all the computers have performed calculation once, the last computer which has performed calculation transmits the calculation result to the first computer which has performed calculation, and after this, calculation is performed by one computer after another and the calculation result is transmitted to the next computer such that the calculation of each cycle is repeated.

According to another aspect of the present invention, there is provided a method of calculating a value of a given function by using an apparatus that includes a plurality of computers, including:

an input process;

an ElGamal cipher text preparation process;

a sequential substitution reencryption process; and

a result output process,

characterized in that the input process includes an information input step of inputting to the plurality of computers information on a circuit including a plurality of gates and information on the plurality of computers, and a dispersion input step of inputting to each of the computers each one of plural pieces of partial data which are obtained by dispersing input data of the function into plural pieces by the number of the computers,

the ElGamal cipher text preparation process includes an ElGamal cipher

text preparation step of generating a set of ElGamal cipher texts in which at least one of the computers corresponds to the gate of the circuit that realizes the given function,

the sequential substitution reencryption process includes a step of allowing each of the computers to perform a substitution reencryption process one after another, and the substitution reencryption process includes a cipher text obtaining step of allowing the computer in this turn to receive the set of ElGamal cipher texts from the computer in the previous turn, a cipher text substitution and reencryption step of changing an order of the set of cipher texts received in the cipher text obtaining step for substitution and subjecting those cipher texts to reencryption, and a step of disclosing the data generated in the cipher text substitution and reencryption step to at least the computer in the next order, and

the result output process includes a partial decryption step of deciphering or partially deciphering a part of the cipher texts generated in the cipher text substitution and reencryption step, a decryption step of deciphering a cipher text that enciphers data corresponding to the input to the circuit in the cipher texts generated in the cipher text substitution and reencryption step, and an evaluation step of evaluating an output of the circuit by using the data deciphered in the decryption step and the data partially deciphered in the partial decryption step.

In this case, the following construction may be adopted that the set of ElGamal cipher texts corresponding to each of the gates is a set of ElGamal cipher texts of a secret key generated corresponding to each of the gate by each of the computers, and

a public key used for generating the ElGamal cipher texts is a sum of public keys corresponding to gates for generating two signals input to this gate.

Furthermore, the following construction may be adopted that the input process further includes a step of inputting an area variable of an ElGamal encryption method to each of the computers,

the ElGamal cipher text preparation process further includes a gate secret key generating step of generating a secret key of the ElGamal cipher texts corresponding to each of the gates of the circuit by each of the computers,

each of the computers performs:

a gate public key generating step of generating a gate public key corresponding to the secret key generated in the gate secret key generating step,

a gate public key validity proof generating step of generating a gate public key validity proof for the public key generated in the gate public key generating step,

a gate public key validity proof disclosing step of disclosing the gate public key validity proof generated in the gate public key validity proof generating step,

an input gate secret key generating step of generating a secret key of the ElGamal cipher texts corresponding to a gate where an input is directly made to the circuit of the gates of the circuit,

an input gate public key generating step of generating an input gate public key corresponding to the secret key generated in the input gate secret key generating step,

an input gate public key validity proof generating step of generating a validity proof for the public key generated in the input gate public key generating step,

an input gate public key validity proof disclosing step of disclosing the input public key validity proof generated in the input gate public key validity proof generating step,

a gate public key obtaining step of obtaining gate public keys generated by other respective computers,

a gate public key integration step of integrating the gate public keys obtained in the gate public key obtaining step,

a gate public key encryption step of enciphering the gate secret key

generated by this computer with the gate public key integrated in the gate public key integration step,

a gate secret key cipher text disclosing step of disclosing a gate secret key cipher text generated in the gate public key encryption step,

a gate secret key cipher text validity proof generating step of generating a validity proof for the gate secret key cipher text,

a gate secret key cipher text validity proof disclosing step of disclosing the gate secret key cipher text validity proof generated in the gate secret key cipher text validity proof generating step,

an input cipher text generating step of generating a cipher text corresponding to a part of the input of the circuit input to each of the computers,

an input cipher text validity proof generating step of generating a validity proof for the cipher text corresponding to the part of the input of the circuit generated in the input cipher text generating step,

an input cipher text validity proof disclosing step of disclosing the proof generated in the input cipher text validity proof generating step, and

an output cipher text generating step of generating and disclosing a cipher text corresponding to an output of the gate,

the sequential substitution reencryption process includes:

a gate secret key cipher text substitution and reencryption step of changing an order of a set of the gate secret key cipher texts with one substitution randomly selected on the basis of a predetermined permitted substitution method for reencryption,

an input cipher text substitution and reencryption step of changing an order of a set of the input cipher texts with one substitution randomly selected on the basis of a predetermined permitted substitution method for reencryption,

an output cipher text substitution and reencryption step of changing an order of a set of the output cipher texts with one substitution randomly selected

on the basis of a predetermined permitted substitution method for reencryption, and

a gate secret key cipher text, input cipher text, and output cipher text substitution and reencryption validity proof generating and disclosing step of generating and disclosing validity proofs for the substitution and reencryption performed in the gate secret key cipher text substitution and reencryption step, the input cipher text substitution and reencryption step, and the output cipher text substitution and reencryption step,

the partial decryption step of the result output process includes:

a gate secret key partial decryption step of partially deciphering the gate secret key cipher texts by mutually performing communication and calculation by the computers,

an input cipher text partial decryption step of partially deciphering the input cipher texts by mutually performing communication and calculation by the computers,

an output cipher text partial decryption step of partially deciphering the output cipher texts by mutually performing communication and calculation by the computers, and

a gate secret key, input cipher text, and output cipher text partial decryption step validity proof generating and disclosing step of generating and disclosing the validity proofs for the partial decryption performed in the gate secret key partial decryption step, the input cipher text partial decryption step, and the output cipher text partial decryption step, and

the calculation method further includes a step of verifying various validity proofs disclosed by other computers.

According to the present invention, there is provided a calculation system for evaluating a function, including:

a plurality of computers,

communication means for performing communication with the plurality of computers,

input process means,

ElGamal cipher text preparation means,

sequential substitution reencryption means, and

result output means,

characterized in that the input means inputs information on a circuit whose output is desired to be obtained, information on the plurality of computers, and information on which part of an input to the circuit each of the computers has,

the ElGamal cipher text preparation means prepares ElGamal cipher texts for generating a set of ElGamal cipher texts corresponding to gates of the circuit that realizes the given function,

the sequential substitution reencryption means includes cipher text obtaining means for allowing the computer in this turn to receive the set of ElGamal cipher texts from the computer in the previous turn, cipher text substitution and reencryption means for changing an order of the set of cipher texts received by the cipher text obtaining means for substitution and subjecting those cipher texts to reencryption, and means for disclosing the data generated by the cipher text substitution and reencryption means to at least the computer in the next order, and

the result output means includes partial decryption means for deciphering or partially deciphering a part of the cipher texts generated by the cipher text substitution and reencryption means, decryption means for deciphering encryption related to itself of a cipher text that enciphers data corresponding to the input to the circuit in the cipher texts generated by the cipher text substitution and reencryption means, and evaluation means for evaluating an output of the circuit while using the data deciphered by the decryption means by the plurality of computers and the data partially deciphered by the partial decryption means

by the plurality of computers.

According to another aspect of the present invention, there is provided a calculation system, including a plurality of computers, input means, and output means, in which one of the computers firstly performs calculation and transmits the calculation result to another computer and the another computer which has received the calculation result performs the next calculation such that calculation is performed by one computer after another, and when all the computers have performed calculation once, the last computer which has performed calculation transmits the calculation result to the first computer which has performed calculation, and after this, calculation is performed by one computer after another and the calculation result is transmitted to the next computer such that the calculation of each cycle is repeated,

characterized in that the input means inputs information on a circuit and a part of input bits to the circuit to the computer,

the calculation of the zero-th cycle is performed before the first computer performs the calculation of the first cycle,

the plurality of computers include data obtaining means for obtaining transmitted data used in the calculation of each cycle, validity proof verifying means, signature text verifying means, first computer special calculating means performed by the first computer, random number generating means for performing random number generation, a main calculation calculating means for performing a main calculation, validity proof generating means for proving a validity for a calculation performed in the main calculation, signature means, and data transmission means,

the transmitted data includes data transmitted from other computer, data main body, a validity proof for the data main body, and a signature text,

the signature text includes data including a signature text corresponding to a combination of the data transmitted from the other computer, the data main

body, and the validity proof for the data main body,

the validity proof verifying means verifies a validity proof in the transmitted data

the signature text verifying means verifies the signature text in the transmitted data,

the main calculation calculates the random number generated by the random number generating means,

the signature means generates a signature text for a combination of the transmitted data, the data main body that is the calculation result calculated in the main calculation, and the validity proof generated by the validity proof generating means, and

the data transmission means transmits a combination of the transmitted data, the data main body that is the calculation result calculated in the main calculation, the validity proof generated by the validity proof generating means, and the signature text generated by the signature means.

In this case, the following construction may be adopted that a data main body of the transmitted data and the data main body that is the calculation result calculated in the main calculation include a combination of multiple sequence alignment ElGamal cipher texts on a true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring in the calculation of the first cycle.

Furthermore, the following construction may be adopted that the calculation of each cycle includes calculation means for the first cycle and calculation means of cycles subsequent to the first cycle,

the calculation means generates the combination of the multiple sequence alignment ElGamal cipher texts on the true value group ring and the extended multiple sequence alignment ElGamal cipher texts on the true value group ring with the calculation means of the zero-th cycle and includes reencryption public

key generating means for generating a public key used for reencryption by the calculation means of the first cycle, data conversion means for converting the transmitted data, secret key conversion means, and random number conversion means,

the data conversion means is adapted to convert the combination of the cipher texts that are the data main body with another combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring,

the secret key conversion means converts the secret key used for the combination of the cipher texts that are the calculation result of the data conversion means with a secret key corresponding to the public key generated by the reencryption public key generating means,

the calculation result of the secret key conversion means includes a combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring,

the random number conversion means is adapted to convert a random number used for the combination of the cipher texts that are the calculation results of the data conversion means, and

the calculation result of the random number conversion means includes a combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring.

Furthermore, the following construction may be adopted that the calculation means of the cycles subsequent to the first cycle includes of the calculation means of the second cycles and the calculation means of cycles subsequent to the second cycle,

the data main body of the transmitted data and the data main body calculated in the main calculation include a combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring in the second calculation, and

the calculation means of the second cycles cipher text conversion means for converting the data main body of the transmitted data to generate an ElGamal cipher text or an ellipse curve ElGamal cipher text and partial decryption means for partially deciphering the cipher texts of the data main body of the transmitted data.

Moreover, the calculation means of the cycles subsequent to the second cycle only includes the calculation means of the third cycle,

the calculation means of the third cycle of the main calculation means outputs the transmitted data as it is, and

the validity proof generating means outputs a null string.

According to the method of calculating the function from the large number of inputs, while the ElGamal encryption method is used, the devices including the means for generating a correspondence table of the gate input and output whose values are concealed and the respective inputs subsequently operates so that orders of the correspondence tables including the set of the ElGamal cipher texts are shuffled as shown in Fig. 5. With the adoption of such a construction, it is possible to achieve the object of the present invention by calculating the output of the function with respect to the inputs that are distributed to the respective devices.

Effect of the Invention

The first effect resides in that the calculation amount of the respective devices and the communication amount are remained to be only in proportion to

the number of gates of the circuit, which is efficient.

This is because the correspondence table of the gate input and output is generated with the ElGamal cipher text for each gate, and any random number generating device is not used, thereby making it easier to prove the validity for the calculation on the correspondence table to the third party.

The second effect resides in that the number of times for the respective calculation devices to perform the communication with another calculation device is remained to be a fixed number irrespective of the number of gates of the circuit to be calculated, which is efficient.

This is because it is necessary to perform the operation for making the correspondence relation of the correspondence tables of the input and output of the respective gates unknown to all the calculation devices, but this operation is realized by the operation of shuffling the correspondence relations in orders by the respective calculation devices.

Brief Description of the Drawings

Fig. 1 is a flowchart for describing a prior art of Non-patent Document 1.

Fig. 2 is a drawing showing a relation between a calculation phase and a communication phase in a garbled circuit parallel construction process according to the prior art of Non-patent Document 1 in which calculation devices to which the same numerals are given are described several times but are the identical devices which operate in a different time, and the time flows from the top to the bottom in the drawing.

Fig. 3 is a flowchart for describing a prior art of Non-patent Document 2.

Fig. 4 is a flowchart for describing only one process out of processes performed in parallel plural times in a parallel rank determination process in the flowchart of Fig. 3 that describes the prior art of Non-patent Document 2.

Fig. 5 is a block diagram showing a calculation process flow devised to

solve the problems of the conventional method by the technology proposed in this application.

Fig. 6 is a flowchart showing a specific example of an operation of a best mode for carrying out a first invention.

Fig. 7 is a flowchart showing a detail of the first half of an ElGamal cipher text preparation process in the specific example of the operation according to the best mode for carrying out the first invention.

Fig. 8 is a flowchart showing a detail of the latter half of the ElGamal cipher text preparation process in the specific example of the operation according to the best mode for carrying out the first invention.

Fig. 9 is a flowchart showing a detail of the first half of a sequential substitution reencryption process in the specific example of the operation according to the best mode for carrying out the first invention in which processes of the flowchart shown in a balloon on the right hand side.

Fig. 10 is a flowchart showing a detail of a result output process in the specific example of the operation according to the best mode for carrying out the first invention.

Fig. 11 is a block diagram showing a construction in the specific example of the operation according to the best mode for carrying out the first invention.

Fig. 12 is a block diagram showing a construction of the calculation device constituting the best mode for carrying out the first invention.

Fig. 13 is a flowchart showing a detail of an input process in the specific example of the operation according to the best mode for carrying out the first invention.

Fig. 14 is a block diagram showing a relation among devices according to a second invention.

Fig. 15 is a flowchart for calculation performed by the respective calculation devices of each cycle from first to third cycles according to the

second invention.

Fig. 16 shows a data flow according to the second invention.

Fig. 17 is a flowchart for main calculation of the first cycle.

Fig. 18 is a flowchart for main calculation of the second cycle.

Fig. 19 is a drawing for facilitating understanding of data that is calculated with respect to each gate in the prior art.

Best Mode for Carrying Out the Invention

Next, embodiments of the present invention will be described with reference to the drawings.

First Embodiment

A first embodiment of the present invention will be described with reference to Figs. 6 to 13.

[Preparation]

[System configuration]

As shown in Fig. 11, λ calculation devices 308 respectively include communication means 307. Hereinafter, the calculation devices are called as follows in turn.

$$\left\{u(\alpha)\right\}_{\alpha=1,\ldots,\lambda}$$

The number of the calculation devices, a relation with a subscript character α which corresponds to the respective calculation devices, and the like are referred to as system configuration information.

[Circuit information]

In a description described later, circuit information is input to the respective calculation devices expressed as follows.

$$\left\{u(\alpha)\right\}_{\alpha=1,\ldots,\lambda}$$

This circuit information will be described.

First, a circuit expressed by input circuit information is denoted by f. The circuit f is a circuit including m logical gates. The respective gates are referred to as $G_1,...,G_i,...,G_m$. Herein, the respective gates have 2 inputs and 1 output. If the circuit is not formed by such gates, the respective gates are replaced by an equivalent circuit including a plurality of 2-input 1-output gates. This replacement method is already known, so the description will be omitted. A wiring for transmitting a signal from one gate of the circuit to an input of another gate transmits a signal which corresponds to 0 or 1. An output wiring of $G_{\lceil k \rfloor}$ is denoted by $w_{(k)}$. The number of wirings to be input to the circuit f is set as n, and this is set as $\{w_{(k)}\}_{k=m+1,...,m+n}$. The wiring may be blanched in a midway to be input to two or more gates. The wiring $[k]$ is adapted to transmit the same signal even in the case of blanching, and the blanched wirings are collectively referred to as $w_{(k)}$. Symbols $w_{(1)},...,w_{(l)}$ denote outputs of the circuit f. All of the wirings of the circuit are either a gate output wiring or a circuit input wiring. Thus, the wirings $\{w_{(k)}\}_{k=1,...,m+n}$ are all the wirings.

When the outputs of the gates G_i and G_j are input to the gate G_k, that is, when the wirings $w_{(i)}$ and $w_{(j)}$ are input to the gate G_k, and the wiring w_k is connected as the wiring for the output of G_k, a relation among the output $b_{\lceil i \rfloor}$ of the gate G_i, the output $b_{\lceil j \rfloor}$ of the gate G_j, and the output $b_{\lceil k \rfloor}$ of the gate G_k, is set as follows.

$$b_{\lceil k \rfloor} = b_{\lceil i \rfloor} \circledcirc_{G_{\lceil k \rfloor}} b_{\lceil j \rfloor}$$

Then, the calculation device $u^{(\alpha)}$ has a part of the signal to be input to the circuit f. The calculation device knows the signal propagated to a part of the wirings $\{w_{(k)}\}_{k=m+1,...,m+n}$.

When the number of bits to be input to the circuit f by the calculation device $u^{(\alpha)}$ is l_α, and the inputs of the all the calculation devices are gathered, the inputs constitute all the inputs to the circuit. That is, the following expression

is obtained.

$$\Sigma_{\alpha=1}^{\lambda} I_\alpha = \lambda$$

With regard to k = m+1, ..., m+n, bits that are input to the respective wirings $w_{(k)}$ are set as $b\lceil k \rfloor$ and the respective bits are allocated by I_α each for the calculation device $u^{(\alpha)}$. In other words, the calculation device $u^{(\alpha)}$ determines the set described below.

$$\{b\lceil k \rfloor \in \{0,1\}\} k = m + \Sigma_{\beta=1}^{\alpha-1} I_\beta + 1, \ldots, m + \Sigma_{\beta=1}^{\alpha} I_\beta$$

Even when the allocation of the circuit numbers is changed, the circuit is not changed fundamentally. Therefore, even when the inputs are allocated in the above-mentioned manner, the generality is not impaired.

The m gates $G_1, \ldots, G_m$, the calculation $\odot_{G[k]}$ performed at the respective gates, the wirings $\{w_{(k)}\}_{k=m+1, \ldots, m+n}$ connected to the respective gates, and the allocation of the input wirings $\{I_\alpha\}_{\alpha=, \ldots, \lambda}$ with respect to the calculation devices are referred to as circuit information 300.

[Group used for the calculation]

In this embodiment, the calculation on an ellipse curve is utilized, so this group will be described. However, the group is not necessarily essential for carrying out the present invention. As a substitution of this group, a commutative multiplicative group such as a calculation on a prime field may be used.

Hereinafter, reference symbol E denotes an ellipse curve whose order is a prime factor q, reference symbol E denotes O an infinite distance of the ellipse curve E, and reference symbol G(≠O) denotes a point on the ellipse curve E. The prime factor q is set sufficiently large to be cryptologically secure. A mapping from the point of the ellipse curve E onto Z/qZ is expressed by φ. The mapping φ whose image space is sufficiently large is selected. As an example of the mapping φ, one of the values of the coordinate on the ellipse curve E is used, or the like. When reference symbol h denotes a member and reference symbol

G denotes a point on the ellipse curve E, a point multiplied by h is expressed as [h]G.

[Notation]

A character written on a right shoulder is a superscript but the character is not an order representing a power. In addition, □ represents an exclusive OR of the bit.

[Input processes 203, 312]

When the process is started, as shown in Fig. 6, first of all, an input process 203 is performed. The input process 203 will be described with reference to Figs. 12 and 13 which show the detail of the process.

In an ElGamal cipher text preparation process, by using information public means and public information obtaining means, disclosure and obtainment of data are both performed (309).

[Determination of area variables]

The calculation determines E, G, and φ. In addition, a method of using a hash function or the like determines a point H on the ellipse curve which no one can find out the original Z/qZ of in which H=[h]G is satisfied. These values E, H, G, and φ are referred to area variables 301. These values are all stored in the calculation device in advance (Step 100 of Fig. 13).

[Input of circuit information and circuit part input]

Information of the circuit f and system configuration information are input to all of the calculation devices (Step 101 of Fig. 13).

To the respective calculation devices $\left\{u^{(\alpha)}\right\}_{\alpha=1,\ldots,\lambda}$, the following partial input dispersed to the circuit is input (Step 102 of Fig. 13).

$$\left\{br_k \in \{0,1\}\right\} k = m+\Sigma_{\beta=1}^{\alpha-1} l_\beta+1, \ldots, m+\Sigma_{\beta=1}^{\alpha} l_\beta$$

[ElGamal cipher text preparation processes 200, 203]

[Setting of secret key and public key for each gate]

Next, as shown in Fig. 6, an ElGamal cipher text preparation process 200 is performed. The ElGamal cipher text preparation process 200 will be described with reference to Figs. 7 and 8 which show the detail of the process.

The respective calculation devices $u^{(\alpha)}$ uniquely and randomly generate gate secret keys described below for all k=1,...,m+n and all $b \in \{0,1\}$ (Step 103 of Fig. 7).

$$X^{(\alpha)b}{}_{[k]} \in_R E$$

$$z^{(\alpha)} \in_R Z/qZ$$

The respective calculation devices $u^{(\alpha)}$ generate the following data for all k=1,...,m+n and all $b \in \{0,1\}$.

$$x^{(\alpha)b}{}_{[k]} = \Phi(X^{(\alpha)b}{}_{[k]})$$

Then, the respective calculation devices $u^{(\alpha)}$ generate gate public keys for all k=1,...,m+n and all $b \in \{0,1\}$ (Step 104 of Fig. 7).

$$Y^{(\alpha)b}{}_{[k]} = [x^{(\alpha)b}{}_{[k]}]G$$

$$Z^{(\alpha)} = [z^{(\alpha)}]G$$

After that, the respective calculation devices u_α use information public devices to make public the gate public keys generated by themselves (Step 105 of Fig. 7). Hereinafter, in the first embodiment, to make public means that the information public device is used to make something public.

In addition, with respect to each $Y^{(\alpha)b}{}_{[k]}, Z^{(\alpha)}$, while following a method A described separately, the respective calculation devices u_α generate the proof of having knowledge of $x^{(\alpha)b}{}_{[k]}, z^{(\alpha)}$ as the validity proof for the gate public key (Step 106 of Fig. 7) and make the proof public (Step 107 of Fig. 7).

[Setting of input public key]

The respective calculation devices u_α generate input gate secret keys for all of the below.

$$k = m+1+\Sigma_{\beta=1}^{\alpha-1} l_\beta, \ldots, m+\Sigma_{\beta=1}^{\alpha} l_\beta$$

The respective calculation devices u_α use the input $b\ \lceil k \rfloor \in \{0,1\}$ to uniquely and randomly generate input gate secret keys described below (Step 108 of Fig. 7).

$$x^{\sim b\ \lceil k \rfloor}{}_{[k]} \in_R Z/qZ$$

The respective calculation devices u_α generate input gate public keys described below (Step 109 of Fig. 7).

$$Y^{\sim b\ \lceil k \rfloor}{}_{[k]} = [x^{\sim b\ \lceil k \rfloor}{}_{[k]}]G$$
$$Y^{\sim b\ \lceil k \rfloor\ \square 1}{}_{[k]} = H - Y^{\sim b\ \lceil k \rfloor}{}_{[k]}$$

With respect to all $k = m+1+\Sigma_{\beta=1}^{\alpha-1} l_\beta, \ldots, m+\Sigma_{\beta=1}^{\alpha} l_\beta$ and all $b \in \{0,1\}$, the respective calculation devices u_α make $Y^{\sim b}{}_{[k]}$ as the input gate public key of the calculation devices (Step 110 of Fig. 7).

In addition, regarding each k, while following a method B described separately, the respective calculation devices u_α generate the proof of having knowledge described below with respect to $b\ \lceil k \rfloor = 0$ or $b\ \lceil k \rfloor = 1$ as the validity proof for the input gate public key (Step 111 of Fig. 7) and make the proof public (Step 112 of Fig. 7).

$$Y^{\sim b\ \lceil k \rfloor}{}_{[k]} = [x^{\sim b\ \lceil k \rfloor}{}_{[k]}]G \text{ which is meant by } x^{\sim b\ \lceil k \rfloor}{}_{[k]}$$

[Processing related to gate]

All the calculation devices $\{u_\alpha\}$ use the public information obtaining means to obtain the gate public key expresses as follows (Step 113 of Fig. 8).

$$\left\{Y^{(\alpha)b}{}_{[k]}, Z^{(\alpha)}\right\}_{\alpha=1,\ldots,\lambda}$$

With respect to all k=1,...,m and all $b \in \{0,1\}$, the calculation devices $\{u_\alpha\}$ integrate the gate public key as expressed below by themselves (Step 114 of Fig. 8).

$$Y^{b}{}_{[k]} = \Sigma_{\alpha=1}{}^{\lambda} Y^{(\alpha)b}{}_{[k]}$$
$$Z = \Sigma_{\alpha=1}{}^{\lambda} Z^{(\alpha)}$$

With respect to all k=1,...,m and all $b \in \{0,1\}$, all the calculation devices $\{u_\alpha\}$ uniquely and randomly generate the following data.

$$r^{(\alpha)\varepsilon}{}_{k} \in_R Z/qZ$$

Then, with respect to all k=1,...,m and all $b,c,\varepsilon \in \{0,1\}$ k=1, encryption is performed through the ellipse ElGamal encryption method, thereby generating a cipher text of the gate secret key described below (Step 115 of Fig. 8) and making the text public (Step 116 of Fig. 8).

$$(C^{(\alpha)bc\varepsilon}{}_{[k]}, D^{(\alpha)bc\varepsilon}{}_{[k]}) = ([r^{(\alpha)\varepsilon}{}_{[k]}]G, X^{(\alpha)\varepsilon}{}_{[k]} + [r^{(\alpha)\varepsilon}{}_{[k]}](Y^{b}{}_{[i]} + Y^{c}{}_{[j]} + Z))$$

It should be noted that the wirings $w_{[i]}$ and $w_{[j]}$ are input to the gate $G_{[k]}$.

In addition, the proof in that the decryption results of the ellipse ElGamal cipher texts $(C^{(\alpha)bc0}{}_{[k]}, D^{(\alpha)bc0}{}_{[k]})$ with respect to all k and all $b,c \in \{0,1\}$ are identical to each other and the decryption results of the ellipse ElGamal cipher texts $(C^{(\alpha)bc1}{}_{[k]}, D^{(\alpha)bc1}{}_{[k]})$ with respect to all k and all $b,c \in \{0,1\}$ are identical to each other is generated as the validity proof for the gate secret key cipher texts by using a method C described separately (Step 117 of Fig. 8) and the proof is made public (Step 118 of Fig. 8).

With respect to all k=1,...,m and all $b,c,\mu,v,\xi \in \{0,1\}$, all the calculation devices $\{u_\alpha\}$ generate secret key identification data cipher texts described below by themselves (Step 119 of Fig. 8).

$(A^{(0)bc}_{[k]\mu,\nu,\xi}, B^{(0)bc}_{[k]\mu,\nu,\xi}) = (O,[\varepsilon]G)$

$[(C^{(0)\alpha bc}_{[k]\mu,\nu,\xi}, D^{(0)\alpha bc}_{[k]\mu,\nu,\xi})]_{\alpha=1,\ldots,\lambda} = [(C^{(\alpha)bc\ \varepsilon}_{[k]}, D^{(\alpha)bc\ \varepsilon}_{[k]})]_{\alpha=1,\ldots,\lambda}$

wherein $\varepsilon = ((b \square \mu) \odot_G \Gamma_{k]}(c \square \nu)) \square \xi$.

[Processing related to input wiring]

With respect to all k=m+1,...,m+n and all $\varepsilon \in \{0,1\}$, all the calculation devices $\{u_\alpha\}$ uniquely and randomly generate the following data.

$r^{\sim(\alpha)\varepsilon}_k \in_R Z/qZ$

Then, with respect to all k=m+1,...,m+n and all $b,\varepsilon \in \{0,1\}$, all the calculation devices $\{u_\alpha\}$ use the ellipse ElGamal encryption method to generate input cipher texts described below (Step 120 of Fig. 8) and make the texts public (Step 121 of Fig. 8).

$(C^{(\alpha)bc\varepsilon}_{[k]}, D^{(\alpha)bc\varepsilon}_{[k]}) = ([r^{\sim(\alpha)\varepsilon}_{[k]}]G, X^{(\alpha)\varepsilon}_{[k]} + [r^{\sim(\alpha)\varepsilon}_{[k]}](Y^{\sim b}_{[k]} + Z))$

In addition, the proof in that the decryption results of the ellipse ElGamal cipher texts $(C^{(\alpha)bc0}_{[k]}, D^{(\alpha)b0}_{[k]})$ with respect to all k and all $b \in \{0,1\}$ are identical to each other and the decryption results of the ellipse ElGamal cipher texts $(C^{(\alpha)b1}_{[k]}, D^{(\alpha)b1}_{[k]})$ with respect to all k and all $b \in \{0,1\}$ are identical to each other is generated as the validity proof for the gate secret key cipher texts by using a method D described separately (Step 122 of Fig. 8) and the proof is made public (Step 123 of Fig. 8).

With respect to all k=m+1,...,m+n and all $b,\xi \in \{0,1\}$, all the calculation devices $\{u_\alpha\}$ generate input secret key identification data cipher texts described below (Step 124 of Fig. 8).

$(A^{(0)b}_{[k]\xi}, B^{(0)b}_{[k]\xi}) = (O,[\varepsilon]G)\ [(C^{(0)\alpha b}_{[k]\xi}, D^{(0)\alpha b}_{[k]\xi})]_{\alpha=1,\ldots,\lambda} = [(C^{(\alpha)b\ \varepsilon}_{[k]}, D^{(\alpha)b\ \varepsilon}_{[k]})]_{\alpha=1,\ldots,\lambda}$

wherein $\varepsilon = b \square \xi$

[Process related to output wiring]

With respect to all the wirings k=1,...,l and all $b,\varepsilon \in \{0,1\}$, all the calculation devices $\{u_\alpha\}$ generate output cipher texts described below (Step 125 of Fig. 8).

$$(A^{\dagger(0)b_{[k]}\xi}, B^{\dagger(0)b_{[k]}\xi}), \&=\& (O,[\varepsilon]G)$$

wherein $\varepsilon = b \square \xi$

[Sequential substitution reencryption process 201 --- substitution and reencryption of gate cipher texts]

Next, a sequential substitution reencryption process 201 is performed as shown in Fig. 6. The sequential substitution reencryption process 201 will be described with reference to Figs. 9 and 12 which show the detail of the process.

With respect to $\alpha=1,...,\lambda$, the calculation devices perform the following process (denoted by reference numeral 304 of Fig. 12) in turn (Step 126 of Fig. 9). In this process, first of all, the respective calculation devices use the public information obtaining means to obtain necessary data (denoted by reference numeral 310 of Fig. 12) Next, the generated data is made public by using the information public means (denoted by reference numeral 311 of Fig. 12). The order of the λ calculation devices is determined. In order that each of the calculation devices obtains the data, all of the other calculation devices in earlier turn before the calculation device need to finish the data disclosure.

[Cipher text obtaining process]

The following are obtained in Step 151 of Fig. 9.

$$A^{(\alpha-1)bc_{[k]}}\ \mu,\nu,\xi, B^{(\alpha-1)bc_{[k]}}\ \mu,\nu,\xi, C^{(\alpha-1)\beta bc_{[k]}}\ \mu,\nu,\xi, D^{(\alpha-1)\beta bc_{[k]}}\ \mu,\nu,\xi$$

with respect to $k=1,...,m$, all $\beta=1,...,\lambda$, and all $b,c,\mu,\nu,\xi \in \{0,1\}$,

$A^{(\alpha-1)b_{[k]}}\ \xi, B^{(\alpha-1)b_{[k]}}\ \xi, C^{(\alpha-1)\beta b_{[k]}}\ \xi, D^{(\alpha-1)\beta b_{[k]}}\ \xi$ with respect to all $k=m+1,...,m+n$, all $\beta=1,...,\lambda$, and all $b,\xi \in \{0,1\}$, and

$A^{\dagger(\alpha-1)b}{}_{[k]}\ \xi, B^{\dagger(\alpha-1)b}{}_{[k]}\ \xi$ with respect to all k=1,...,l, and all $b,\xi \in \{0,1\}$.

[Wiring signal value and substitution generation]

The calculation devices u_α uniquely and randomly generate substitutions $\{\pi(k) \in_R \{0,1\}\}_{k=1,\ldots,m+n}$ of the signal values of the respective wirings (Step 127 of Fig. 9).

[Random number generation for reencryption]

The calculation devices u_α uniquely and randomly generate random numbers described below which are used for gate secret key reencryption (Step 128 of Fig. 9).

$\{s^{(\alpha)bc}{}_{[k]}\ \mu, \nu, \xi\}_{k=1,\ldots,m;b,c,\mu,\nu,\xi \in R\{0,1\}}$

$\{t^{(\alpha)\beta bc}{}_{[k]}\ \mu, \nu, \xi\}_{k=1,\ldots,m;\beta=1,\ldots,\beta;b,c,\mu,\nu,\xi \in R\{0,1\}}$

$\{s^{(\alpha)b}{}_{[k]}\ \xi\}_{k=m+1,\ldots,m+n;b,\xi \in R\{0,1\}}$

$\{t^{(\alpha)\beta b}{}_{[k]}\ \xi\}_{k=m+1,\ldots,m+n;\beta=1,\ldots,\lambda;b,\xi \in R\{0,1\}}$

$\{s^{\dagger b}{}_{[k]}\ \xi\}_{k=1,\ldots,l;b,\xi \in R\{0,1\}}$

[Substitution and reencryption of gate secret key cipher texts]

With respect to all k=1,...,m and all $b,c,\mu,\nu,\xi \in \{0,1\}$, the following data is generated by substituting the gate secret key for reencryption (Step 129 of Fig. 9).

$$A^{(\alpha)bc}[k]_{\mu,\nu,\xi} = A^{(\alpha-1)bc}[k]_{\mu\,\square\,\pi(i),\nu\,\square\,\pi(j),\xi\,\square\,\pi(k)} + [s^{(\alpha)bc}[k]_{\mu,\nu,\xi}]G$$

$$B^{(\alpha)bc}[k]_{\mu,\nu,\xi} = B^{(\alpha-1)bc}[k]_{\mu\,\square\,\pi(i),\nu\,\square\,\pi(j),\xi\,\square\,\pi(k)} + [s^{(\alpha)bc}[k]_{\mu,\nu,\xi}](Y^{b}[i]+Y^{c}[j]+Z)$$

$$C^{(\alpha)\beta\,bc}[k]_{\mu,\nu,\xi} = C^{(\alpha-1)\beta\,bc}[k]_{\mu\,\square\,\pi(i),\nu\,\square\,\pi(j),\xi\,\square\,\pi(k)} + [t^{(\alpha)\beta\,bc}[k]_{\mu,\nu,\xi}]G$$

$$D^{(\alpha)\beta\,bc}[k]_{\mu,\nu,\xi} = D^{(\alpha-1)\beta\,bc}[k]_{\mu\,\square\,\pi(i),\nu\,\square\,\pi(j),\xi\,\square\,\pi(k)} + [t^{(\alpha)\beta\,bc}[k]_{\mu,\nu,\xi}](Y^{b}[i]+Y^{c}[j]+Z)$$

[Substitution and reencryption of input cipher texts]

With respect to all k=m+1,...,m+n, all β=1,...,λ, and all $b,\xi \in \{0,1\}$, the following data is generated by substituting the input cipher texts for reencryption (Step 130 of Fig. 9).

$$A^{(\alpha)b}[k]_{\xi} = A^{(\alpha-1)b}[k]_{\xi\,\square\,\pi(k)} + [s^{(\alpha)b}[k]_{\xi}]G$$

$$B^{(\alpha)b}[k]_{\xi} = B^{(\alpha-1)\,b}[k]_{\xi\,\square\,\pi(k)} + [s^{(\alpha)b}[k]_{\xi}](Y^{\wedge b}[k]+Z)$$

$$C^{(\alpha)\beta\,b}[k]_{\xi} = C^{(\alpha-1)\beta\,b,sub>}[k]_{\xi\,\square\,\pi(k)} + [t^{(\alpha)b}[k]_{\xi}]G$$

$$D^{(\alpha)\beta\,b}[k]_{\xi} = D^{(\alpha-1)\beta\,b}[k]_{\xi\,\square\,\pi(k)} + [t^{(\alpha)b}[k]_{\xi}](Y^{\wedge b}[k]+Z)$$

[Substitution and reencryption of output cipher texts]

With respect to all k=1,...,l and all $b,\xi \in \{0,1\}$, the following data is generated by substituting the output cipher texts for reencryption (Step 131 of Fig. 9).

$$A^{\dagger(\alpha)b}[k]_{\xi} = A^{\dagger(\alpha-1)b}[k]_{\xi\,\square\,\pi(k)} + [s^{\dagger b}[k]_{\xi}]G$$

$$B^{\dagger(\alpha)b}[k]_{\xi} = B^{\dagger(\alpha-1)b}[k]_{\xi\,\square\,\pi(k)} + [s^{\dagger b}[k]_{\xi}](Y^{b}[k]+Z)$$

[Validity proof for substitution and reencryption]

$\{A^{(\alpha)bc}{}_{[k]\ \mu,\nu,\xi}, B^{(\alpha)bc}{}_{[k]\ \mu,\nu,\xi}, C^{(\alpha)\beta bc}{}_{[k]\ \mu,\nu,\xi}, D^{(\alpha)\beta bc}{}_{[k]\ \mu,\nu,\xi}\}$

with respect to all $k=1,...,m$, all $\beta=1,...,\lambda$, and all $b,c,\mu,\nu,\xi \in \{0,1\}$,

$\{A^{(\alpha)b}{}_{[k]\ \xi}, B^{(\alpha)b}{}_{[k]\ \xi}, C^{(\alpha)\beta b}{}_{[k]\ \xi}, D^{(\alpha)\beta b}{}_{[k]\ \xi}\}$ with respect to all

$k=m+1,...,m+n$, all $\beta=1,...,\lambda$, and all $b,\xi \in \{0,1\}$, and $A^{\dagger(\alpha)b}{}_{[k]\ \xi}, B^{\dagger(\alpha)b}{}_{[k]\ \xi}$

with respect to all $k=1,...,l$ and all $b,\xi \in \{0,1\}$ are transmitted to the calculation device $u_{\alpha+1}$.

In addition, the proof in that the above-mentioned process is appropriately performed is generated as the validity proof for the substitution and the reencryption regarding the gate secret key cipher texts, the input cipher texts, and the output cipher texts in accordance with a method E described separately and the proof is made public (Step 132 of Fig. 9).

[Result output processes 202, 305]

Next, as shown in Fig. 6, a result output process 202 is performed. The process will be described with reference to Figs. 10 to 12 which show the detail of the process.

In the result output process 202, the information public means and public information obtaining means are used to perform both the disclosure and obtainment of the data (Step 312 of Fig. 12). Finally, the outputs of their own circuits are output (denoted by reference numeral 313 of Fig. 12).

[Partial decryption of gate cipher texts]

All the calculation devices $\{u_\alpha\}_{\alpha=1,...,\lambda}$ partially decipher the gate secret key as described below with respect to all $k=1,...,m, b,c \in \{0,1\}, \beta=1,...,\lambda$ and make the result public (Step 134 of Fig. 10).

$A^{\ddagger(\lambda)\alpha bc}{}_{[k]000} = [z^{(\alpha)}]A^{(\lambda)bc}{}_{[k]000}$

$C^{\ddagger(\lambda)\alpha\beta bc}{}_{[k]000} = [z^{(\alpha)}]C^{(\lambda)\beta bc}{}_{[k]000}$

All the calculation devices $\{u_\alpha\}_{\alpha=1,\ldots,\lambda}$ partially decipher the input cipher texts as described below with respect to all $k = m+1,\ldots,m+n, b \in \{0,1\}, \beta = 1,\ldots,\lambda$ and make the result public (Step 135 of Fig. 10).

$A^{\ddagger(\lambda)\alpha b}{}_{[k]0} = [z^{(\alpha)}]A^{(\lambda)b}{}_{[k]0}$

$C^{\ddagger(\lambda)\alpha\beta b}{}_{[k]0} = [z^{(\alpha)}]C^{(\lambda)\beta b}{}_{[k]0}$

All the calculation devices $\{u_\alpha\}_{\alpha=1,\ldots,\lambda}$ partially decipher the output cipher texts as described below with respect to all $k = 1,\ldots,l, b \in \{0,1\}$ and make the result public (Step 136 of Fig. 10).

$A^{\dagger\ddagger(\lambda)\alpha b}{}_{[k]0} = [z^{(\alpha)}]A^{\dagger(\lambda)b}{}_{[k]0}$

In addition, the proof in that the above-mentioned process is appropriately performed is generated as the validity proof for the partial decryption regarding the gate secret key cipher texts, the input cipher texts, and the output cipher texts in accordance with a method F described separately and the proof is made public (Step 137 of Fig. 10).

[Gate cipher text generation]

Furthermore, all the calculation devices generate the following data as the gate cipher texts with respect to all $k=1,\ldots,m$, all $\alpha=1,\ldots,\lambda$, and all $b,c, \in \{0,1\}$.

$A^{bc}{}_{[k]} = A^{(\lambda)bc}{}_{[k]000}$

$B^{bc}{}_{[k]} = B^{(\lambda)bc}{}_{[k]000} - \Sigma_{\alpha=1}{}^{\lambda} A^{\ddagger(\lambda)\alpha bc}{}_{[k]000}$

$C^{\alpha bc}{}_{[k]} = C^{(\lambda)\alpha bc}{}_{[k]000}$

$D^{\alpha bc}{}_{[k]} = D^{(\lambda)\alpha bc}{}_{[k]000} - \Sigma_{\alpha=1}{}^{\lambda} C^{\ddagger(\lambda)\alpha\beta bc}{}_{[k]000}$

All the calculation devices generate the following data as the gate cipher texts with respect to all $k=m+1,\ldots,m+n$, all $\alpha=1,\ldots,\lambda$, and all $b \in \{0,1\}$.

$$A^{b}_{[k]} = A^{(\lambda)b}_{[k]0}$$

$$B^{b}_{[k]} = B^{(\lambda)b}_{[k]0} - \Sigma_{\alpha=1}^{\lambda} A\#^{(\lambda)\alpha b}_{[k]0}$$

$$C^{\alpha b}_{[k]} = C^{(\lambda)\alpha b}_{[k]0}$$

$$D^{\alpha b}_{[k]} = D^{(\lambda)\alpha b}_{[k]0} - \Sigma_{\alpha=1}^{\lambda} C\#^{(\lambda)\alpha\beta b}_{[k]0}$$

Then, all the calculation devices generate the following data as the gate cipher texts with respect to all k=1,...,l and all $b \in \{0,1\}$ (Step 138 of Fig. 10).

$$A\dagger^{b}_{[k]} = A\dagger^{(\lambda)b}_{[k]0}$$

$$B\dagger^{b}_{[k]} = B\dagger^{(\lambda)b}_{[k]0} - A\dagger\#^{(\lambda)\alpha b}_{[k]0}$$

[Decryption of inputs]

The calculation devices $u^{(\alpha)}$ generate the following data with respect to all

$k = m+1+\Sigma_{\gamma=1}^{\alpha-1} l_\gamma, \ldots, m+\Sigma_{\gamma=1}^{\alpha} l_\gamma$ and all $\beta=1,\ldots,\lambda$ without disclosing $b^{\lceil k \rfloor}$ and make the data public (Step 139 of Fig. 10).

$$G^{b}_{[k]} = B^{b\lceil k \rfloor}_{[k]} - [x\tilde{}^{b\lceil k \rfloor}_{[k]}]A^{b\lceil k \rfloor}_{[k]}$$

$$X^{b(\beta)}_{[k]} = D^{\beta b\lceil k \rfloor}_{[k]} - [x\tilde{}^{b\lceil k \rfloor}_{[k]}]C^{\beta b\lceil k \rfloor}_{[k]}$$

$$x^{b(\beta)}_{[k]} = \phi(X^{b(\beta)}_{[k]})$$

The data made public is referred to as data deciphered from the input cipher texts.

All calculation devices u_α checks the following data with respect to all k=m+1,...,m+n and all $\beta=1,\ldots,\lambda$ to check the validity of the input cipher text decryption (Step 140 of Fig. 10).

$Y^{(\beta)\varepsilon\ \Gamma k\rfloor}{}_{[k]} = [x^{b(\beta)}{}_{[k]}]G$

wherein if $G^{b}[k] = 0$, $\varepsilon_k = 0$, and if $G^{b}[k]=G$, $\varepsilon_k = 1$.

[Circuit evaluation]

With respect to all the gates $G_{kk=1,...,m}$, all the calculation devices u_α find out outputs on the basis of the inputs in an appropriate order in the following manner (Step 141 of Fig. 10). This is the process for the circuit evaluation. It should be noted that outputs of the gates G_i and G_j are input to the gate G_k.

The following data is obtained with respect to $\beta=1,...,\lambda$ (Step 142 of Fig. 10).

$$G^{b}{}_{[k]}=B^{b\Gamma i\rfloor b\Gamma j\rfloor}{}_{[k]}-[\Sigma_{\gamma=1}\lambda(x(\gamma)b\Gamma i\rfloor_{[k]}+x(\gamma)b\Gamma j\rfloor_{[k]})]\,A^{b\Gamma i\rfloor b\Gamma j\rfloor}{}_{[k]}$$

$$X^{b(\beta)}{}_{[k]}=D^{\beta}\,{}_{b\Gamma i\rfloor\,b\Gamma j\rfloor}{}_{[k]}-[\Sigma_{\gamma=1}\lambda(x(\gamma)b\Gamma i\rfloor_{[k]}+x(\gamma)b\Gamma j\rfloor_{[k]})]\,C^{\beta}\,{}_{b\Gamma i\rfloor\,b\Gamma j\rfloor}{}_{[k]}$$

$$x^{b(\beta)}{}_{[k]}=\phi(X^{b(\beta)}{}_{[k]})$$

With respect to all $\beta=1,...,\lambda$, $Y^{(\beta)\varepsilon\ \Gamma k\rfloor}{}_{[k]} = [x^{b(\beta)}{}_{[k]}]G$ is checked (Step 143 of Fig. 10). It should be noted that if $G^{b}[k] = 0$, $\varepsilon_k = 0$, and if $G^{b}[k]=G$, $\varepsilon_k = 1$.

With respect to k to which the above-mentioned process is performed, $b\Gamma k\rfloor = \varepsilon_k$ is set.

[Output evaluation]

The proofs made public up to this process are verified by a verifier (Step 144 of Fig. 10). If the verifier accepts all the proof texts, in other words, if no violations are found, the following process for the output deciphering and disclosure is performed.

With respect to k=1,...,l, all the calculation devices u_α obtain the following data (Step 145 of Fig. 10).

$$G^{\dagger}{}_{[k]}=B^{\dagger b\Gamma k\rfloor}{}_{[k]}-[\Sigma_{\gamma=1}\lambda(x(\gamma)b\Gamma k\rfloor_{[k]}]\,A^{\dagger b\Gamma k\rfloor}{}_{[k]}$$

With respect to k=1,...,l, all the calculation devices $\{u_\alpha\}_{\alpha=1,\ldots,\lambda}$ generate the following data (Step 146 of Fig. 10).

$$A^{\dagger\ddagger}{}_{[k]} = [z^{(\alpha)}]A^{\dagger b\ \lceil k \rfloor}{}_{[k]}$$

Then, the data is made public (Step 147 of Fig. 10).

In addition, the validity proof for this calculation is generated as the validity proof for the output deciphering in accordance with a method G described separately (Step 148 of Fig. 10) and the validity proof is made public (Step 149 of Fig. 10).

The respective calculation devices u_α find out a circuit output (denoted by reference numeral 306 of Fig. 11) on the basis of the following data (Step 150 of Fig. 10).

$$G^{\flat}{}_{[k]} = G^{\dagger}{}_{[k]} - \Sigma_{\gamma=1}{}^{\lambda} A^{\dagger\ddagger}{}_{[k]}$$

With respect to k=1,...,l, if $G^{\flat}{}_{[k]} = 0$, $b\ \lceil k \rfloor = 0$, and if $G^{\flat}{}_{[k]} = G$, $b\ \lceil k \rfloor = 1$.

[Separately described processes]

[Separate description A]

With respect to all k=1,...,m+n and $b \in \{0,1\}$, the certifier (calculator) u_α uniquely and randomly generate the following data.

$$x'^{(\alpha)b}{}_{[k]} \in_R Z \setminus qZ$$

$$z'^{(\alpha)}{}_{[k]} \in_R Z \setminus qZ$$

$$Y'^{(\alpha)b}{}_{[k]} \in [x'^{(\alpha)b}{}_{[k]}]G$$

$$Z'^{(\alpha)} = [z'^{(\alpha)}]G$$

Further, the following data is generated.

$$\theta = \mathrm{Hash}(E,G,\{Y'^{(\alpha)b}{}_{[k]}\}_{k=1,\ldots,m+n;b=0,1}, Z'^{(\alpha)}, \{Y'^{(\alpha)b}{}_{[k]}\}_{k=1,\ldots,m+n;b=0,1}, Z'^{(\alpha)}) \bmod q$$

Then, with respect to all k=1,...,m+n and $b \in \{0,1\}$, the following data is generated.

$$x''^{(\alpha)}{}_{b[k]} = \theta\ x^{(\alpha)}{}_{b[k]} + x'^{(\alpha)}{}_{b[k]} \bmod q$$

$$z''^{(\alpha)} = \theta\ z^{(\alpha)} + z'^{(\alpha)} \bmod q$$

The certifier sets the following data as the proof with respect to all $k=1,\ldots,m+n$ and $b \in \{0,1\}$.

$$Y'^{(\alpha)}{}_{b[k]}, Z'^{(\alpha)}, z^{(\alpha)}{}_{b[k]}, z'^{(\alpha)}$$

The verification method for the above-mentioned proof is as follows. The verifier calculates the first expression to check the second and third expressions described below.

$$\theta = \text{Hash}(E,G,\{Y^{(\alpha)}{}_{b[k]}\}_{k=1,\ldots,m+n;b=0,1}, Z^{(\alpha)}, \{Y'^{(\alpha)}{}_{b[k]}\}_{k=1,\ldots,m+n;b=0,1}, Z'^{(\alpha)}) \bmod q$$

$$[x''^{(\alpha)}{}_{b[k]}]G = [\theta]\, Y^{(\alpha)}{}_{b[k]} + Y'^{(\alpha)}{}_{b[k]}$$

$$[z''^{(\alpha)}]G = \theta\ Z^{(\alpha)} + Z'^{(\alpha)}$$

[Separate description B]

Each certifier (calculator) $u^{(\alpha)}$ uniquely and randomly generates the following data with respect to all $k = m+1+\Sigma_{\beta=1}^{\alpha-1} l_\beta, \ldots, m+\Sigma_{\beta=1}^{\alpha} l_\beta$ for the selected $b\ \Gamma k \lrcorner \in \{0,1\}$.

$$\tilde{x}'^{b\ \Gamma k \lrcorner}{}_{[k]} \in_R Z/qZ$$

Then, the following data is generated.

$$\tilde{Y}'^{b\ \Gamma k \lrcorner}{}_{[k]} = [\tilde{x}'^{b\ \Gamma k \lrcorner}{}_{[k]}]G$$

Moreover, the following data is randomly generated.

$$\theta^{b\Gamma k \lrcorner \square 1}{}_{[k]} \in_R Z/qZ$$

$$\tilde{x}''^{b\Gamma k \lrcorner \square 1}{}_{[k]} \in_R Z/qZ$$

After that, the following data is randomly generated.

$$Y\tilde{}^{b\lceil k\rfloor\square 1}{}_{[k]} = [x\tilde{}^{b\lceil k\rfloor\square 1}{}_{[k]}]G - [\theta^{b\lceil k\rfloor\square 1}{}_{[k]}]\, Y\tilde{}^{b\lceil k\rfloor\square 1}{}_{[k]}$$

The certifier u_α generates the following data with respect to all

$k = m+1+\Sigma_{\beta=1}{}^{\alpha-1}l_\beta,\ldots,m+\Sigma_{\beta=1}{}^{\alpha}l_\beta$.

$$\theta_{[k]} = \mathrm{Hash}(E,G,\{Y\tilde{}^{b}_{[k]}, Y\tilde{}^{b}_{[k]}\}_{b=0,1}) \bmod q$$

$$\theta^{b\lceil k\rfloor}{}_{[k]} = \theta_{[k]} - \theta^{b\lceil k\rfloor\square 1}{}_{[k]} \bmod q$$

Furthermore the following data is generated.

$$x\tilde{}^{b\lceil k\rfloor}{}_{[k]} = \theta^{b\lceil k\rfloor}{}_{[k]}\, x\tilde{}^{b\lceil k\rfloor}{}_{[k]} + x\tilde{}^{b\lceil k\rfloor}{}_{[k]} \bmod q$$

The certifier u_α sets the following data as the proof with respect to all

$k = m+1+\Sigma_{\beta=1}{}^{\alpha-1}l_\beta,\ldots,m+\Sigma_{\beta=1}{}^{\alpha}l_\beta$, $b = 0,1$.

$$Y\tilde{}^{b}_{[k]}, \theta^{0}_{[k]}, x\tilde{}^{b}_{[k]}$$

The verification method for the above-mentioned proof is as follows.

With respect to all $k = m+1+\Sigma_{\beta=1}{}^{\alpha-1}l_\beta,\ldots,m+\Sigma_{\beta=1}{}^{\alpha}l_\beta$, $b = 0,1$, the verifier generates the following data.

$$\theta_{[k]} = \mathrm{Hash}(E,G,[Y\tilde{}^{b}_{[k]}, Y\tilde{}^{b}_{[k]}]_{b=0,1}) \bmod q$$

$$\theta^{1}_{[k]} = \theta_{[k]} - \theta^{0}_{[k]} \bmod q$$

With respect to all

$$k=m+1+\Sigma_{\beta=1}{}^{\alpha-1}l_\beta,\ldots,m+\Sigma_{\beta=1}{}^{\alpha}l_\beta,\ b=0,1$$

it is checked whether the following expressions are established.

$$[\tilde{x}''^{b}_{[k]}]G = [\theta^{b}_{[k]}]\,\tilde{Y}^{b}_{[k]} + \tilde{Y}''^{b}_{[k]}$$

$$\tilde{Y}^{0}_{[k]} + \tilde{Y}^{1}_{[k]} = H$$

[Separate description C]

The certifier (calculator) u_α uniquely and randomly generates the following data with respect to all k=1,...,m and all ε=0,1.

$r'^{(\alpha)\varepsilon 0}_{[k]} \in Z / qZ$

Then, the following data is generated.

$$F^{(\alpha)\varepsilon 0}_{[k]} = [r'^{(\alpha)\varepsilon 0}_{[k]}]G$$

$$F^{(\alpha)\varepsilon 1}_{[k]} = [r'^{(\alpha)\varepsilon 1}_{[k]}]\,(Y^{1}_{[i]} - Y^{0}_{[i]})$$

$$F^{(\alpha)\varepsilon 2}_{[k]} = [r'^{(\alpha)\varepsilon 2}_{[k]}]\,(Y^{1}_{[j]} - Y^{0}_{[j]})$$

Furthermore, the following data is generated.

$$\theta^{(\alpha)}_{[k]} = \mathrm{Hash}\,(E,G,\{C^{(\alpha)bc\,\varepsilon}_{[k]}, D^{(\alpha)bc\,\varepsilon}_{[k]}\}_{k=1,\ldots,m;b,c,\varepsilon=0,1}, \{F^{(\alpha)\varepsilon 0}_{[k]}, F^{(\alpha)\varepsilon 1}_{[k]}, F^{(\alpha)\varepsilon 2}_{[k]},\}_{k=1,\ldots,m;\varepsilon=0,1})$$

Next, the following data is generated.

$$r''^{(\alpha)\varepsilon}_{[k]} = \theta^{(\alpha)}_{[k]}\, r^{(\alpha)\varepsilon}_{[k]} + r'^{(\alpha)\varepsilon}_{[k]} \bmod q$$

The certifier sets the following data as the proof for k=1,...,m and ε=0,1.

$$F^{(\alpha)\varepsilon 0}_{[k]},\ F^{(\alpha)\varepsilon 1}_{[k]},\ F^{(\alpha)\varepsilon 2}_{[k]},\ r''^{(\alpha)\varepsilon}_{[k]}$$

The verification method for the above-mentioned proof is as follows.

The verifier first checks whether the $C^{(\alpha)bc\varepsilon}_{[k]}$ are the same value with

respect to each of ε=0,1 and k=1,...,m for all b,c=0,1.

Next, the following data is generated.

$$\theta(\alpha)_{[k]}, = \text{Hash}\,(E,G,[\,C(\alpha)bc\ \varepsilon_{[k]}, D(\alpha)bc\ \varepsilon_{[k]}\,]_{k=1,\ldots,m;b,c,\varepsilon=0,1}, [\,F(\alpha)\varepsilon 0_{[k]}, F(\alpha)\varepsilon 1_{[k]}, F(\alpha)\varepsilon 2_{[k]},]_{k=1,\ldots,m;\varepsilon=0,1})$$

Next, it is checked whether or not the following expressions are established with respect to all k=1,...,m and ε=0,1.

$$[r'(\alpha)\varepsilon_{[k]}]G = [\theta(\alpha)_{[k]}]\,C(\alpha)00\,\varepsilon_{[k]} + F(\alpha)\varepsilon 0_{[k]}$$

$$[r'(\alpha)\ \varepsilon_{[k]}]\,(Y^1_{[j]} - Y^0_{[j]}) = [\theta(\alpha)_{[k]}]\,(D(\alpha)01\ \varepsilon_{[k]} - D(\alpha)00\,\varepsilon_{[k]}) + F(\alpha)\varepsilon 2_{[k]}$$

$$[r'(\alpha)\ \varepsilon_{[k]}]\,(Y^1_{[i]} - Y^0_{[i]}) = [\theta(\alpha)_{[k]}]\,(D(\alpha)10\ \varepsilon_{[k]} - D(\alpha)00\,\varepsilon_{[k]}) + F(\alpha)\varepsilon 1_{[k]}$$

$$(D(\alpha)11\ \varepsilon_{[k]} - D(\alpha)10\ \varepsilon_{[k]}) = (D(\alpha)01\ \varepsilon_{[k]} - D(\alpha)00\,\varepsilon_{[k]})$$

[Separate description D]

With respect to all k=m+1,...,m+n and all b,ε=0,1, the certifier (calculator) uα uniquely and randomly generates $r'(\alpha)b_{[k]} \in Z/qZ$. Then, the following data is generated.

$$F(\alpha)\varepsilon 0_{[k]} = [r''(\alpha)\,\varepsilon_{[k]}]G$$

$$F(\alpha)\varepsilon 1_{[k]} = [r''(\alpha)\,\varepsilon_{[k]}]\,(Y''1_{[k]} - Y''0_{[k]})$$

Furthermore, with respect to all k=m+1,...,m+n, the following data is generated.

$$\theta(\alpha)_{[k]}, = \text{Hash}\,(E,G,[\,C(\alpha)b\,\varepsilon_{[k]}, D(\alpha)b\,\varepsilon_{[k]}, F(\alpha)b\,\varepsilon_{[k]}\,]_{k=m+1,\ldots,m+n;b,\varepsilon=0,1})$$

Next, with respect to all k=m+1,...,m+n and all b,ε=0,1, the following data is generated.

$$r''(\alpha)\,\varepsilon_{[k]} = \theta(\alpha)_{[k]}\,r''(\alpha)\,\varepsilon_{[k]} + r''(\alpha)\,\varepsilon_{[k]} \bmod q$$

The certifier sets the following data as the proof with respect to k=m+1,...,m+n and ε=0,1.

$$F(\alpha)\varepsilon 0_{[k]},\ F(\alpha)\varepsilon 1_{[k]},\ r''(\alpha)\varepsilon_{[k]}$$

The verification method for the above-mentioned proof is as follows. First of all, the verifier checks whether or not the values of $C^{(\alpha)bc}_{(k)}$ are the same with respect to each of ε=0,1 and k=1,...,m for all b=0,1

Next, the following data is generated.

$$\theta(\alpha)_{[k]},\ = \text{Hash}\ (E,G,\ \{\ C(\alpha)b\varepsilon_{[k]},\ D(\alpha)b\varepsilon_{[k]},\ F(\alpha)b\varepsilon_{[k]}\ \}_{k=m+1,...,m+n;b,\varepsilon=0,1}\ )$$

Then, it is checked whether or not the following expressions are established with respect to all k=m+1,...,m+n and all ε=0,1.

$$[r''(\alpha)\varepsilon_{[k]}]G = [\theta(\alpha)_{[k]}]\ C(\alpha)0\varepsilon_{[k]} + F(\alpha)\varepsilon 0_{[k]}$$

$$[r''(\alpha)\ \varepsilon_{[k]}]\ (Y^{\wedge}1_{[j]} - Y^{\wedge}0_{[j]}) = [\theta(\alpha)_{[k]}]\ (D(\alpha)1\ \varepsilon_{[k]} - D(\alpha)0\varepsilon_{[k]}) + F(\alpha)\varepsilon 1_{[k]}$$

[Separate description E]

With respect to α=1,...,λ, the calculator u_α performs the following process in turn.

The calculator u_α performs uniquely and randomly selects the following data from Z/qZ.

$$\{\sigma(\alpha-h)bc_{[k]}\mu,\nu,\xi,\ \tau(\alpha-h)\beta bc_{[k]}\mu,\nu,\xi,\}_{k=1,...,m;\beta=1,...,\lambda;h=2/3,1/3,0;b,c,\mu,\nu,\xi\in\{0,1\}}$$

wherein

$$\Sigma_{h=0,1/3,2/3}\ \sigma(\alpha-h)bc_{[k]}\mu,\nu,\xi = s(\alpha)bc_{[k]}\mu,\nu,\xi \mod q$$

$$\Sigma_{h=0,1/3,2/3}\ \tau(\alpha-h)\beta bc_{[k]}\mu,\nu,\xi = t(\alpha)\beta bc_{[k]}\mu,\nu,\xi \mod q$$

The calculator u_α generates the following data with respect to all k=1,...,m, and all $b,c,\mu,\nu,\xi \in \{0,1\}$.

$$A^{(\alpha-2/3)bc}[k]_{\mu,\nu,\xi} = A^{(\alpha-1)bc}[k]_{\mu \square \pi(i),\nu,\xi} + [\sigma^{(\alpha-2/3)bc}[k]_{\mu,\nu,\xi}]G$$

$$B^{(\alpha-2/3)bc}[k]_{\mu,\nu,\xi} = B^{(\alpha-1)bc}[k]_{\mu \square \pi(i),\nu,\xi} + [\sigma^{(\alpha-2/3)bc}[k]_{\mu,\nu,\xi}](Y^b[i]+Y^c[j]+Z)$$

$$C^{(\alpha-2/3)\beta\ bc}[k]_{\mu,\nu,\xi} = C^{(\alpha-1)\beta\ bc}[k]_{\mu \square \pi(i),\nu\ \xi} + [\tau^{(\alpha-2/3)\beta\ bc}[k]_{\mu,\nu,\xi}]G$$

$$D^{(\alpha-2/3)\beta\ bc}[k]_{\mu,\nu,\xi} = D^{(\alpha-1)\beta\ bc}[k]_{\mu \square \pi(i),\nu,\xi} + [\tau^{(\alpha-2/3)\beta\ bc}[k]_{\mu,\nu,\xi}](Y^b[i]+Y^c[j]+Z)$$

$$A^{(\alpha-1/3)bc}[k]_{\mu,\nu,\xi} = A^{(\alpha-2/3)bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\sigma^{(\alpha-1/3)bc}[k]_{\mu,\nu,\xi}]G$$

$$B^{(\alpha-1/3)bc}[k]_{\mu,\nu,\xi} = B^{(\alpha-2/3)bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\sigma^{(\alpha-1/3)bc}[k]_{\mu,\nu,\xi}](Y^b[i]+Y^c[j]+Z)$$

$$C^{(\alpha-1/3)\beta\ bc}[k]_{\mu,\nu,\xi} = C^{(\alpha-2/3)\beta\ bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\tau^{(\alpha-1/3)\beta\ bc}[k]_{\mu,\nu,\xi}]G$$

$$D^{(\alpha-1/3)\beta\ bc}[k]_{\mu,\nu,\xi} = D^{(\alpha-2/3)\beta\ bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\tau^{(\alpha-1/3)\beta\ bc}[k]_{\mu,\nu,\xi}](Y^b[i]+Y^c[j]+Z)$$

$$A^{(\alpha)bc}[k]_{\mu,\nu,\xi} = A^{(\alpha-1/3)bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\sigma^{(\alpha)bc}[k]_{\mu,\nu,\xi}]G$$

$$B^{(\alpha)bc}[k]_{\mu,\nu,\xi} = B^{(\alpha-1/3)bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\sigma^{(\alpha)bc}[k]_{\mu,\nu,\xi}](Y^b[i]+Y^c[j]+Z)$$

$$C^{(\alpha)\beta\ bc}[k]_{\mu,\nu,\xi} = C^{(\alpha-1/3)\beta\ bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\tau^{(\alpha)\beta\ bc}[k]_{\mu,\nu,\xi}]G$$

$$D^{(\alpha)\beta\ bc}[k]_{\mu,\nu,\xi} = D^{(\alpha-1/3)\beta\ bc}[k]_{\mu,\nu \square \pi(j),\xi} + [\tau^{(\alpha)\beta\ bc}[k]_{\mu,\nu,\xi}](Y^b[i]+Y^c[j]+Z)$$

Next, the calculator u_α uniquely and randomly selects the following data from Z/qZ with respect to all k=1,...,m, all β=1, all h=2/3,1/3,0, and all $b,c,\mu,\nu,\xi \in \{0,1\}$.

$$\sigma^{(\alpha-h)bc}[k]_{\mu,\nu,\xi},\ \tau^{(\alpha-h)\beta\ bc}[k]_{\mu,\nu,\xi},$$

Next, the calculator u_α generates the following data with respect to all k=1,...,m, all β=1, and all $b,c,\mu,\nu,\xi \in \{0,1\}$.

$A'^{(\alpha-2/3)bc}[k]\pi(i),\mu,\nu,\xi = A^{(\alpha-1)bc}[k]\mu \square \pi(i),\nu,\xi + [\sigma'^{(\alpha-2/3)bc}[k]\mu,\nu,\xi]G$

$B'^{(\alpha-2/3)bc}[k]\pi(i):\mu,\nu,\xi = B^{(\alpha-1)bc}[k]\mu \square \pi(i),\nu,\xi + [\sigma'^{(\alpha-2/3)bc}[k]\mu,\nu,\xi](Y^b[i]+Y^c[j]+Z)$

$C'^{(\alpha-2/3)\beta\ bc}[k]\pi(i),\mu,\nu,\xi = C^{(\alpha-1)\beta\ bc}[k]\mu \square \pi(i),\nu\ \xi + [\tau'^{(\alpha-2/3)\beta\ bc}[k]\mu,\nu,\xi]G$

$D'^{(\alpha-2/3)\beta\ bc}[k]\pi(i),\mu,\nu,\xi = D^{(\alpha-1)\beta\ bc}[k]\mu \square \pi(i),\nu,\xi + [\tau'^{(\alpha-2/3)\beta\ bc}[k]\mu,\nu,\xi](Y^b[i]+Y^c[j]+Z)$

$A'^{(\alpha-1/3)bc}[k]\pi(j)\mu,\nu,\xi = A^{(\alpha-2/3)bc}[k]\mu,\nu \square \pi(j),\xi + [\sigma'^{(\alpha-1/3)bc}[k]\mu,\nu,\xi]G$

$B'^{(\alpha-1/3)bc}[k]\pi(j)\mu,\nu,\xi = B^{(\alpha-2/3)bc}[k]\mu,\nu \square \pi(j),\xi + [\sigma'^{(\alpha-1/3)bc}[k]\mu,\nu,\xi](Y^b[i]+Y^c[j]+Z)$

$C'^{(\alpha-1/3)\beta\ bc}[k]\pi(j)\mu,\nu,\xi = C^{(\alpha-2/3)\beta\ bc}[k]\mu,\nu \square \pi(j),\xi + [\tau'^{(\alpha-1/3)\ \beta\ bc}[k]\mu,\nu,\xi]G$

$D'^{(\alpha-1/3)\beta\ bc}[k]\pi(j)\mu,\nu,\xi = D^{(\alpha-2/3)\beta\ bc}[k]\mu,\nu \square \pi(j),\xi + [\tau'^{(\alpha-1/3)\ \beta\ bc}[k]\mu,\nu,\xi](Y^b[i]+Y^c[j]+Z)$

$A'^{(\alpha)bc}[k]\pi(k)\mu,\nu,\xi = A^{(\alpha-1/3)bc}[k]\mu,\nu \square \pi(j),\xi + [\sigma'^{(\alpha)bc}[k]\mu,\nu,\xi]G$

$B'^{(\alpha)bc}[k]\pi(k)\mu,\nu,\xi = B^{(\alpha-1/3)bc}[k]\mu,\nu \square \pi(j),\xi + [\sigma'^{(\alpha)bc}[k]\mu,\nu,\xi](Y^b[i]+Y^c[j]+Z)$

$C'^{(\alpha)\beta\ bc}[k]\pi(k)\mu,\nu,\xi = C^{(\alpha-1/3)\beta\ bc}[k]\mu,\nu \square \pi(j),\xi + [\tau'^{(\alpha)\beta\ bc}[k]\mu,\nu,\xi]G$

$D'^{(\alpha)\beta\ bc}[k]\pi(k)\mu\,nu\,\xi = D^{(\alpha-1/3)\beta\ bc}[k]\mu,\nu \square \pi(j),\xi + [\tau'^{(\alpha)\ \beta\ bc}[k]\mu,\nu,\xi](Y^b[i]+Y^c[j]+Z)$

Furthermore, with respect to all k=m+1,...,m+n, all β=1,...,λ, and all $b,\xi \in \{0,1\}$, $s'^{(\alpha)b}{}_{(k)\xi}, t'^{(\alpha)b}{}_{(k)\xi}$ is uniquely and randomly selected from Z/qZ.

With respect to all k=m+1,...,m+n and all $b,\xi \in \{0,1\}$, the following data is generated.

$$A'^{(\alpha)b}_{[k]\pi(k),\xi} = A^{(\alpha-1)b}_{[k]\xi} \square \pi(k) + [s'^{(\alpha)b}_{[k]\xi}]G$$

$$B'^{(\alpha)b}_{[k]\pi(k),\xi} = B^{(\alpha-1)b}_{[k]\xi} \square \pi(k) + [s'^{(\alpha)b}_{[k]\xi}](Y^{b}_{[k]}+Z)$$

$$C'^{(\alpha)\beta b}_{[k]\pi(k)\xi} = C^{(\alpha-1)\beta b}_{[k]\xi} \square \pi(k) + [t'^{(\alpha)\beta b}_{[k]\xi}]G$$

$$D'^{(\alpha)\beta b}_{[k]\pi(k)\xi} = D^{(\alpha-1)\beta b}_{[k]\xi} \square \pi(k) + [t'^{(\alpha)\beta b}_{[k]\xi}](Y^{b}_{[k]}+Z)$$

Then, the following data is uniquely and randomly selected from Z/qZ with respect to all k=1,...,l and all $b,\xi \in \{0,1\}$.

$$s^{\dagger\prime(\alpha)b}_{[k]\xi}$$

With respect to all k=1,...,l and all $b,\xi \in \{0,1\}$, the following data is generated.

$$A^{\dagger\prime(\alpha)b}_{[k]\pi(k)\xi} = A^{\dagger(\alpha-1)b}_{[k]\xi} \square \pi(k) + [s^{\dagger\prime(\alpha)b}_{[k]\xi}]G$$

$$B^{\dagger\prime(\alpha)b}_{[k]\pi(k)\xi} = B^{\dagger(\alpha-1)b}_{[k]\xi} \square \pi(k) + [s^{\dagger\prime(\alpha)b}_{[k]\xi}](Y^{b}_{[k]}+Z)$$

With respect to all k=1,...,m and all h=2/3,1/3,0, the following data is uniquely and randomly selected from Z/qZ.

$$\theta^{(\alpha)}_{[k]\pi(i)\square 1}$$

With respect to all k=1,...,m, all β=1,...,λ, and all $b,c,\mu,\nu,\xi \in \{0,1\}$, the following data is uniquely and randomly selected from Z/qZ.

$$\sigma''^{(\alpha-2/3)bc}_{[k]\ \pi(i)\square 1,\mu,\nu,\xi},\ \sigma''^{(\alpha-1/3)bc}_{[k]\ \pi(j)\square 1,\mu,\nu,\xi},\ \sigma''^{(\alpha)bc}_{[k]\ \pi(k)\square 1,\mu,\nu,\xi},\ \tau''^{(\alpha-2/3)\beta\ bc}_{[k]\ \pi(i)\square 1,\mu,\nu,\xi},\ \tau''^{(\alpha-1/3)\beta\ bc}_{[k]\ \pi(j)\square 1,\mu,\nu,\xi},\ \tau''^{(\alpha)\beta bc}_{[k]\ \pi(k)\square 1,\mu,\nu,\xi}$$

Next, with respect to all k=m+1,...,m+n, all β=1,...,λ, and all $b,\xi \in \{0,1\}$,

the following data is uniquely and randomly selected from Z/qZ.

$$s''(\alpha)b_{[k]}\ \pi(k)\square 1,\xi\ ,\ t''(\alpha)\beta b_{[k]}\ \pi(k)\square 1,\xi$$

Next, with respect to all k=1,...,l and all $b,\xi \in \{0,1\}$, the following data is uniquely and randomly selected from Z/qZ.

$$s^{\dagger''(\alpha)b}{}_{[k]}\ \pi(k)\square 1,\xi\ ,\ t^{\dagger''(\alpha)b}{}_{[k]}\ \pi(k)\square 1,\xi$$

Next, with respect to all k=1,...,m, all β=1,...,λ, and all $b,c,\mu,\nu,\xi \in \{0,1\}$, the following data is generated.

$A'^{(\alpha-2/3)bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi \doteq [\sigma''^{(\alpha-2/3)bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(i)\square 1]\ A^{(\alpha-1)bc}_{[k]}\ \mu\square\pi(i),\nu,\xi$

$B'^{(\alpha-2/3)bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi = [\sigma''^{(\alpha-2/3)bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi]\ (Y^{b}_{[i]}+Y^{c}_{[j]}+Z) - [\theta^{(\alpha)}_{[k]}\pi(i)\square 1]\ B^{(\alpha-1)bc}_{[k]}\ \mu\square\pi(i),\nu,\xi$

$C'^{(\alpha-2/3)\beta\ bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi = [\tau''^{(\alpha-2/3)\beta\ bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(i)\square 1]\ C^{(\alpha-1)\beta\ bc}_{[k]}\ \mu\square\pi(i),\nu,\xi$

$D'^{(\alpha-2/3)\beta\ bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi = [\tau''^{(\alpha-2/3)\beta\ bc}_{[k]}\ \pi(i)\square 1,\mu,\nu,\xi]\ (Y^{b}_{[i]}+Y^{c}_{[j]}+Z) - [\theta^{(\alpha)}_{[k]}\pi(i)\square 1]\ D^{(\alpha-1)\beta\ bc}_{[k]}\ \mu\square\pi(i),\nu,\xi$

$A'^{(\alpha-1/3)bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi = [\sigma''^{(\alpha-1/3)bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(j)\square 1]\ A^{(\alpha-2/3)bc}_{[k]}\ \mu,\nu\square\pi(j),\xi$

$B'^{(\alpha-1/3)bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi = [\sigma''^{(\alpha-1/3)bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi]\ (Y^{b}_{[i]}+Y^{c}_{[j]}+Z) - [\theta^{(\alpha)}_{[k]}\pi(j)\square 1]\ B^{(\alpha-2/3)bc}_{[k]}\ \mu,\nu\square\pi(j),\xi$

$C'^{(\alpha-1/3)\beta\ bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi = [\tau''^{(\alpha-1/3)\beta\ bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(j)\square 1]\ C^{(\alpha-2/3)\beta\ bc}_{[k]}\ \mu,\nu\square\pi(j),\xi$

$D'^{(\alpha-1/3)\beta\ bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi = [\tau''^{(\alpha-1/3)\beta\ bc}_{[k]}\ \pi(j)\square 1,\mu,\nu,\xi]\ (Y^{b}_{[i]}+Y^{c}_{[j]}+Z) - [\theta^{(\alpha)}_{[k]}\pi(j)\square 1]\ D^{(\alpha-2/3)\beta\ bc}_{[k]}\ \mu,\nu\square\pi(j),\xi$

$A'^{(\alpha)bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi = [\sigma''^{(\alpha)bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(k)\square 1]\ A^{(\alpha-1/3)bc}_{[k]}\ \mu,\nu,\xi\square\pi(k)$

$B'^{(\alpha)bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi = [\sigma''^{(\alpha)bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi]\ (Y^{b}_{[i]}+Y^{c}_{[j]}+Z) - [\theta^{(\alpha)}_{[k]}\pi(k)\square 1]\ B^{(\alpha-1/3)bc}_{[k]}\ \mu,\nu,\xi\square\pi(k)$

$C'^{(\alpha)\beta\ bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi = [\tau''^{(\alpha)\beta\ bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(k)\square 1]\ C^{(\alpha-1/3)\beta\ bc}_{[k]}\ \mu,\nu,\xi\square\pi(k)$

$D'^{(\alpha)\beta\ bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi = [\tau''^{(\alpha)\beta\ bc}_{[k]}\ \pi(k)\square 1,\mu,\nu,\xi]\ (Y^{b}_{[i]}+Y^{c}_{[j]}+Z) - [\theta^{(\alpha)}_{[k]}\pi(k)\square 1]\ D^{(\alpha-1/3)\beta\ bc}_{[k]}\ \mu,\nu,\xi\square\pi(k)$

Next, with respect to all k=m+1,...,m+n, all β=1,...,λ, and all $b,\xi \in [0,1)$, the following data is generated.

$A'^{(\alpha)b}_{[k]}\ \pi(i)\Box 1,\xi = [s''^{(\alpha-1)b}_{[k]}\ \pi(i)\Box 1,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(i)\Box 1]\ A^{(\alpha-1)b}_{[k]}\ \xi\Box\pi(i)$

$B'^{(\alpha)b}_{[k]}\ \pi(i)\Box 1,\xi = [s''^{(\alpha-1)b}_{[k]}\ \pi(i)\Box 1,\xi](Y^{-b}_{[k]} + Z) - [\theta^{(\alpha)}_{[k]}\pi(i)\Box 1]\ B^{(\alpha-1)b}_{[k]}\ \xi\Box\pi(i)$

$C'^{(\alpha)\beta b}_{[k]}\ \pi(i)\Box 1,\xi = [t''^{(\alpha-1)\beta b}_{[k]}\ \pi(i)\Box 1,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(i)\Box 1]\ C^{(\alpha-1)\beta b}_{[k]}\ \xi\Box\pi(i)$

$D'^{(\alpha)\beta b}_{[k]}\ \pi(i)\Box 1,\xi = [t''^{(\alpha-1)\beta b}_{[k]}\ \pi(i)\Box 1,\xi](Y^{-b}_{[k]} + Z) - [\theta^{(\alpha)}_{[k]}\pi(i)\Box 1]\ D^{(\alpha-1)\beta b}_{[k]}\ \xi\Box\pi(i)$

Next, with respect to all k=1,...,l and all $b,\xi \in \{0,1\}$, the following data is generated.

$A^{\dagger\prime(\alpha)b}_{[k]}\ \pi(i)\Box 1,\xi = [s^{\dagger\prime\prime(\alpha-1)b}_{[k]}\ \pi(i)\Box 1,\xi]G - [\theta^{(\alpha)}_{[k]}\pi(i)\Box 1]\ A^{\dagger(\alpha-1)b}_{[k]}\ \xi\Box\pi(i)$

$B^{\dagger\prime(\alpha)b}_{[k]}\ \pi(i)\Box 1,\xi = [s^{\dagger\prime\prime(\alpha-1)b}_{[k]}\ \pi(i)\Box 1,\xi](Y^{b}_{[k]} + Z) - [\theta^{(\alpha)}_{[k]}\pi(i)\Box 1]\ B^{\dagger(\alpha-1)b}_{[k]}\ \xi\Box\pi(i)$

Next, the following data is generated.

S = { E,G,

{

$A^{(\alpha-h)bc}_{[k]\mu,\nu,\xi}$, $B^{(\alpha-h)bc}_{[k]\mu,\nu,\xi}$, $C^{(\alpha-h)\beta bc}_{[k]\mu,\nu,\xi}$, $D^{(\alpha-h)\beta bc}_{[k]\mu,\nu,\xi}$,

$A'^{(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}$, $B'^{(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}$, $C'^{(\alpha-h)\beta bc}_{[k]\zeta,\mu,\nu,\xi}$, $D'^{(\alpha-h)\beta bc}_{[k]\zeta,\mu,\nu,\xi}$

$k=1,...,m; h=2/3,1/3,0; \beta=1,...,\lambda$; $b,c,\mu,\nu,\xi,\zeta \in \{0,1\}$

{

$A^{(\alpha)b}_{[k]\xi}$, $B^{(\alpha)b}_{[k]\xi}$, $C^{(\alpha)\beta b}_{[k]\xi}$, $D^{(\alpha)\beta b}_{[k]\xi}$,

$A'^{(\alpha)b}_{[k]\zeta,\xi}$, $B'^{(\alpha)b}_{[k]\zeta,\xi}$, $C'^{(\alpha)\beta b}_{[k]\zeta,\xi}$, $D'^{(\alpha)\beta b}_{[k]\zeta,\xi}$

$k=m+1,...,m+n; \beta=1,...,\lambda$; $b,\xi,\zeta \in \{0,1\}$ }

{

$A^{\dagger(\alpha)b}_{[k]\xi}$, $B^{\dagger(\alpha)b}_{[k]\xi}$

$A^{\dagger'(\alpha)b}_{[k]\zeta,\xi}$, $B^{\dagger'(\alpha)b}_{[k]\zeta,\xi}$

$k=1,...,l; b,\xi,\zeta \in \{0,1\}$ }

Next, each verifier uα generates the following data with respect to all k=1,...,m+n.

$$\theta^{(\alpha)}_{[k]} = \text{Hash}(E,G,k,S)$$

Next, with respect to all k=1,...,m+n, the following data is generated.

$$\theta^{(\alpha)}_{[k]\pi(i)} = \theta^{(\alpha)}_{[k]} - \theta^{(\alpha)}_{[k]\pi(i)}\square 1$$

Next, with respect to all k=1,...,m, all β=1,...,λ, all h=2/3,1/3,0, and all $b,c,\mu,\nu,\xi \in \{0,1\}$, the following data is generated.

$\sigma''^{(\alpha-h)bc}_{[k]\pi(i),\mu,\nu,\xi} = \theta^{(\alpha)}_{[k]\pi(i)}\ \sigma'^{(\alpha-2/3)bc}_{[k]\mu,\nu,\xi} + \sigma'^{(\alpha-2/3)bc}_{[k]\mu,\nu,\xi} \bmod q$

$\tau''^{(\alpha-h)\beta bc}_{[k]\pi(i),\mu,\nu,\xi} = \theta^{(\alpha)}_{[k]\pi(i)}\ \tau'^{(\alpha-2/3)\beta\ bc}_{[k]\mu,\nu,\xi} + \tau'^{(\alpha-2/3)\beta\ bc}_{[k]\mu,\nu,\xi} \bmod q$

Next, with respect to all k=m+1,...,m+n, all β=1,...,λ, and all $b,\xi \in \{0,1\}$, the following data is generated.

$s''^{(\alpha)b}_{[k]\pi(i)\xi} = \theta^{(\alpha)}_{[k]\pi(i)}\ s'^{(\alpha-1)b}_{[k]\xi} + s'^{(\alpha-1)b}_{[k]\xi} \bmod q$

$t''^{(\alpha)\beta b}_{[k]\pi(i)\xi} = \theta^{(\alpha)}_{[k]\pi(i)}\ t'^{(\alpha-1)\beta b}_{[k]\xi} + t'^{(\alpha-1)b}_{[k]\xi} \bmod q$

Next, with respect to all k=1,...,l and $b,\xi \in \{0,1\}$, the following data is generated.

$s\dagger''^{(\alpha)b}_{[k]\pi(i)\xi} = \theta^{(\alpha)}_{[k]\pi(i)}\ s\dagger'^{(\alpha-1)b}_{[k]\xi} + s\dagger'^{(\alpha-1)b}_{[k]\xi} \bmod q$

Finally, the following data is set as proofs.

$A^{(\alpha-h)bc}_{[k]\mu,\nu,\xi}, B^{(\alpha-h)bc}_{[k]\mu,\nu,\xi}, C^{(\alpha-h)\beta bc}_{[k]\mu,\nu,\xi}, D^{(\alpha-h)\beta bc}_{[k]\mu,\nu,\xi},$

$A'^{(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}, B'^{(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}, C'^{(\alpha-h)\beta bc}_{[k]\zeta,\mu,\nu,\xi}, D'^{(\alpha-h)\beta bc}_{[k]\zeta,\mu,\nu,\xi},$

$\sigma''^{(\alpha)b}_{[k]\zeta,\xi}, \tau''^{(\alpha)\beta b}_{[k]\zeta,\xi},$

with respect to all $k=1,...,m;h=2/3,1/3,0;\beta=1,...,\lambda;b,c,\mu,\nu,\xi,\zeta \in \{0,1\}$.

$A^{(\alpha)b}_{[k]\xi}$, $B^{(\alpha)b}_{[k]\xi}$, $C^{(\alpha)\beta b}_{[k]\xi}$, $D^{(\alpha)\beta b}_{[k]\xi}$,

$A'^{(\alpha)b}_{[k]\zeta,\xi}$, $B'^{(\alpha)b}_{[k]\zeta,\xi}$, $C'^{(\alpha)\beta b}_{[k]\zeta,\xi}$, $D'^{(\alpha)\beta b}_{[k]\zeta,\xi}$,

$s''^{(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}$, $t''^{(\alpha-h)\beta bc}_{[k]\zeta,\mu,\nu,\xi}$,

with respect to all $k=m+1,...,m+n;\beta=1,...,\lambda;b,\xi,\zeta\in\{0,1\}$.

$A^{\dagger(\alpha)b}_{[k]\xi}$, $B^{\dagger(\alpha)b}_{[k]\xi}$,

$A^{\dagger'(\alpha)b}_{[k]\zeta,\xi}$, $B^{\dagger'(\alpha)b}_{[k]\zeta,\xi}$,

$s^{\dagger''(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}$,

with respect to all $k=1,...,l;b,\xi,\zeta\in\{0,1\}$.

$\theta^{(\alpha)}_{[k]\zeta}$

with respect to all $k=1,...,m+n$.

The verification method for the above-mentioned proof is as follows.

The verifier generates the following data.

$S = [E,G,$

$[$

$A^{(\alpha-h)bc}_{[k]\mu,\nu,\xi}, B^{(\alpha-h)bc}_{[k]\mu,\nu,\xi}, C^{(\alpha-h)\beta bc}_{[k]\mu,\nu,\xi}, D^{(\alpha-h)\beta bc}_{[k]\mu,\nu,\xi},$

$A'^{(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}, B'^{(\alpha-h)bc}_{[k]\zeta,\mu,\nu,\xi}, C'^{(\alpha-h)\beta bc}_{[k]\zeta,\mu,\nu,\xi}, D'^{(\alpha-h)\beta bc}_{[k]\zeta,\mu,\nu,\xi}$

$]_{k=1,\ldots,m;h=2/3,1/3,0;\beta=1,\ldots,\lambda;b,c,\mu,\nu,\xi,\zeta\in\{0,1\}}$

$[$

$A^{(\alpha)b}_{[k]\xi}, B^{(\alpha)b}_{[k]\xi}, C^{(\alpha)\beta b}_{[k]\xi}, D^{(\alpha)\beta b}_{[k]\xi},$

$A'^{(\alpha)b}_{[k]\zeta,\xi}, B'^{(\alpha)b}_{[k]\zeta,\xi}, C'^{(\alpha)\beta b}_{[k]\zeta,\xi}, D'^{(\alpha)\beta b}_{[k]\zeta,\xi}$

$]_{k=m+1,\ldots,m+n;\beta=1,\ldots,\lambda;b,\xi,\zeta\in\{0,1\}}$

$[$

$A^{\dagger(\alpha)b}_{[k]\xi}, B^{\dagger(\alpha)b}_{[k]\xi}$

$A^{\dagger'(\alpha)b}_{[k]\zeta,\xi}, B^{\dagger'(\alpha)b}_{[k]\zeta,\xi}$

$]_{k=1,\ldots,l;b,\xi,\zeta\in\{0,1\}}]$

Each verifier uα generates the following data with respect to all k=1,...,m+n.

$\theta^{(\alpha)}_{[k]}$ = Hash(E,G,k, S)

It is checked whether or not the following expression is established with respect to all k=1,...,m.

$$\theta^{(\alpha)}_{[k]0} + \theta^{(\alpha)}_{[k]1} = \theta^{(\alpha)}_{[k]}$$

Next, with respect to

$k=1,\ldots,m;h=2/3,1/3,0;\beta=1,\ldots,\lambda;b,c,\mu,\nu,\xi,\zeta\in\{0,1\}$, the

verifier checks whether the following expressions are established.

$$[\sigma''^{(\alpha-2/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}]G = [\theta^{(\alpha)}{}_{[k]\zeta}](A^{(\alpha-2/3)bc}{}_{[k]\mu,\nu,\xi} - A^{(\alpha-1)bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - A'^{(\alpha-2/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\sigma''^{(\alpha-2/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}](Y^{b}{}_{[i]}+Y^{c}{}_{[j]}+Z) = [\theta^{(\alpha)}{}_{[k]\zeta}](B^{(\alpha-2/3)bc}{}_{[k]\mu,\nu,\xi} - B^{(\alpha-1)bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - B'^{(\alpha-2/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\tau''^{(\alpha-2/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}]G = [\theta^{(\alpha)}{}_{[k]\zeta}](C^{(\alpha-2/3)\beta\ bc}{}_{[k]\mu,\nu,\xi} - C^{(\alpha-1)\beta\,bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - C'^{(\alpha-2/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\tau''^{(\alpha-2/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}](Y^{b}{}_{[i]}+Y^{c}{}_{[j]}+Z) = [\theta^{(\alpha)}{}_{[k]\zeta}](D^{(\alpha-2/3)\beta\ bc}{}_{[k]\mu,\nu,\xi} - D^{(\alpha-1)\beta\,bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - D'^{(\alpha-2/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\sigma''^{(\alpha-1/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}]G = [\theta^{(\alpha)}{}_{[k]\zeta}](A^{(\alpha-1/3)bc}{}_{[k]\mu,\nu,\xi} - A^{(\alpha-2/3)bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - A'^{(\alpha-1/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\sigma''^{(\alpha-1/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}](Y^{b}{}_{[i]}+Y^{c}{}_{[j]}+Z) = [\theta^{(\alpha)}{}_{[k]\zeta}](B^{(\alpha-2/3)bc}{}_{[k]\mu,\nu,\xi} - B^{(\alpha-2/3)bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - B'^{(\alpha-1/3)bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\tau''^{(\alpha-1/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}]G = [\theta^{(\alpha)}{}_{[k]\zeta}](C^{(\alpha-1/3)\beta\ bc}{}_{[k]\mu,\nu,\xi} - C^{(\alpha-2/3)\beta\ bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - C'^{(\alpha-1/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\tau''^{(\alpha-1/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}](Y^{b}{}_{[i]}+Y^{c}{}_{[j]}+Z) = [\theta^{(\alpha)}{}_{[k]\zeta}](D^{(\alpha-1/3)\beta\ bc}{}_{[k]\mu,\nu,\xi} - D^{(\alpha-2/3)\beta\ bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - D'^{(\alpha-1/3)\beta\ bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\sigma''^{(\alpha)bc}{}_{[k]\zeta,\mu,\nu,\xi}]G = [\theta^{(\alpha)}{}_{[k]\zeta}](A^{(\alpha)bc}{}_{[k]\mu,\nu,\xi} - A^{(\alpha-1/3)bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - A'^{(\alpha)bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\sigma''^{(\alpha)bc}{}_{[k]\zeta,\mu,\nu,\xi}](Y^{b}{}_{[i]}+Y^{c}{}_{[j]}+Z) = [\theta^{(\alpha)}{}_{[k]\zeta}](B^{(\alpha-1/3)bc}{}_{[k]\mu,\nu,\xi} - B^{(\alpha-2/3)bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - B'^{(\alpha)bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\tau''^{(\alpha)\beta\,bc}{}_{[k]\zeta,\mu,\nu,\xi}]G = [\theta^{(\alpha)}{}_{[k]\zeta}](C^{(\alpha)\beta\,bc}{}_{[k]\mu,\nu,\xi} - C^{(\alpha-1/3)\beta\ bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - C'^{(\alpha)\beta\,bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

$$[\tau''^{(\alpha)\beta\,bc}{}_{[k]\zeta,\mu,\nu,\xi}](Y^{b}{}_{[i]}+Y^{c}{}_{[j]}+Z) = [\theta^{(\alpha)}{}_{[k]\zeta}](D^{(\alpha)\beta\,bc}{}_{[k]\mu,\nu,\xi} - D^{(\alpha-1/3)\beta\ bc}{}_{[k]\mu\square\zeta,\nu,\xi}) - D'^{(\alpha)\beta\,bc}{}_{[k]\zeta,\mu,\nu,\xi}$$

Next, with respect to $k=m+1,\ldots,m+n;\beta=1,\ldots,\lambda;b,\xi,\zeta\in\{0,1\}$, the verifier checks whether or not the following expressions are established.

$$[s''(\alpha)b_{[k]}\zeta,\xi]G=[\theta(\alpha)_{[k]}\zeta](A(\alpha)b_{[k]}\xi-A(\alpha-1)b_{[k]}\xi\square\zeta)-A'(\alpha)b_{[k]}\zeta,\xi$$

$$[s''(\alpha)b_{[k]}\zeta,\xi](Y''b_{[k]}+Z)=[\theta(\alpha)_{[k]}\zeta](B(\alpha)b_{[k]}\xi-B(\alpha-1)b_{[k]}\xi\square\zeta)-B'(\alpha)b_{[k]}\zeta,\xi$$

$$[t''(\alpha)\beta b_{[k]}\zeta,\xi]G=[\theta(\alpha)_{[k]}\zeta](C(\alpha)\beta b_{[k]}\xi-C(\alpha-1)\beta b_{[k]}\xi\square\zeta)-C'(\alpha)\beta b_{[k]}\zeta,\xi$$

$$[t''(\alpha)\beta b_{[k]}\zeta,\xi](Y''b_{[k]}+Z)=[\theta(\alpha)_{[k]}\zeta](D(\alpha)\beta b_{[k]}\xi-D(\alpha-1)\beta b_{[k]}\xi\square\zeta)-D'(\alpha)\beta b_{[k]}\zeta,\xi$$

Next, with respect to $k=1,\ldots,l;b,\xi,\zeta\in\{0,1\}$, the verifier checks whether or not the following expressions are established.

$$[s\dagger''(\alpha)b_{[k]}\zeta,\xi]G=[\theta(\alpha)_{[k]}\zeta](A\dagger(\alpha)b_{[k]}\xi-A\dagger(\alpha-1)b_{[k]}\xi\square\zeta)-A\dagger'(\alpha)b_{[k]}\zeta,\xi$$

$$[s\dagger''(\alpha)b_{[k]}\zeta,\xi](Yb_{[k]}+Z)=[\theta(\alpha)_{[k]}\zeta](B\dagger(\alpha)b_{[k]}\xi-B\dagger(\alpha-1)b_{[k]}\xi\square\zeta)-B\dagger'(\alpha)b_{[k]}\zeta,\xi$$

[Separate description F]

Regarding the certifier (calculator), with respect to all k=1,...,m+n and all $b,c\in\{0,1\}$, all the certifiers (calculators) $\{u_\alpha\}_{\alpha=1,\ldots,\lambda}$ uniquely and randomly generate $z'(\alpha)\in_R Z/qZ$ and generate $Z'(\alpha)=[z'(\alpha)]G$. With respect to all k=1,...,m, all $b,c\in\{0,1\}$, and all β=1,...,λ, the following date is generated.

$$A'(\lambda)\alpha bc_{[k]}000=[z'(\alpha)]A(\lambda)bc_{[k]}000$$

$$C'(\lambda)\alpha\beta bc_{[k]}000=[z'(\alpha)]C(\lambda)\alpha bc_{[k]}000$$

With respect to all k=m+1,...,m+n, all $b \in [0,1]$, and all β=1,...,λ, the following date is generated.

$$A'(\lambda)\alpha b_{[k]0} = [z'(\alpha)]A(\lambda)b_{[k]0}$$

$$C'(\lambda)\alpha\beta b_{[k]0} = [z'(\alpha)]C(\lambda)\alpha b_{[k]0}$$

With respect to all k=1,...,l and all $b \in [0,1]$, the following date is generated.

$$A\dagger'(\lambda)\alpha b_{[k]0} = [z'(\alpha)]A\dagger(\lambda)b_{[k]0}$$

Then, the following date is generated.

$$S = [\{A^{(\lambda)bc}_{[k]000}, C^{(\lambda)\beta bc}_{[k]000}\}_{k=1,\ldots,m;b,c=0,1;\beta=1,\ldots,\lambda},$$

$$\{A^{(\lambda)b}_{[k]0}, C^{(\lambda)\beta b}_{[k]0}\}_{k=m+1,\ldots,m+n;b=0,1;\beta=1,\ldots,\lambda},$$

$$\{A^{\dagger(\lambda)b}_{[k]0}\}_{k=1,\ldots,l;b=0,1},$$

$$\{A^{\sharp(\lambda)\alpha bc}_{[k]000}, C^{\sharp(\lambda)\alpha\beta bc}_{[k]000}\}_{k=1,\ldots,m;b,c=0,1;\beta=1,\ldots,\lambda},$$

$$\{A^{\sharp(\lambda)\alpha b}_{[k]0}, C^{\sharp(\lambda)\alpha \text{ beta } b}_{[k]0} k=m+1,\ldots,m+n;b=0,1;\beta=1,\ldots,\lambda,$$

$$\{A^{\dagger\sharp(\lambda)\alpha b}_{[k]0}\}_{k=1,\ldots,l;b=0,1},$$

$$\{z'(\alpha)\},$$

$$\{A'^{(\lambda)\alpha bc}_{[k]000}, C'^{(\lambda)\alpha\beta bc}_{[k]000}\}_{k=1,\ldots,m;b,c=0,1;\beta=1,\ldots,\lambda},$$

$$\{A'^{(\lambda)\alpha b}_{[k]0}, C'^{(\lambda)\alpha\beta b}_{[k]0}\}_{k=m+1,\ldots,m+n;b=0,1;\beta=1,\ldots,\lambda},$$

$$\{A^{\dagger\prime(\lambda)\alpha b}_{[k]0}\}_{k=1,\ldots,l;b=0,1}, .]\}$$

Furthermore, the following date is generated.

$$\theta = \text{Hash}(E,G,S) \bmod q$$

In addition, the following date is generated.

$$z''(\alpha) = z(\alpha)\theta + z'(\alpha) \bmod q$$

The certifier sets the following data as the proof.

$Z'(\alpha)$, $\{A'(\lambda)^{\alpha bc}_{[k]000}, C'(\lambda)^{\alpha\beta bc}_{(k)000}\}_{k=1,...,m;b,c=0,1;\beta=1,...,\lambda}$,

$\{A'(\lambda)^{\alpha b}_{[k]0}, C'(\lambda)^{\alpha\beta b}_{[k]0}\}_{k=m+1,...,m+n;b=0,1;\beta=1,...,\lambda}$,

$\{A\dagger'(\lambda)^{\alpha b}_{[k]0}\}_{k=1,...,l;b=0,1}$,

$z''(\alpha)$

The verification method for the above-mentioned proof is as follows.

The verifier generates the following data,

$S = [[A(\lambda)^{bc}_{[k]000}, C(\lambda)^{\beta bc}_{[k]000}]_{k=1,...,m;b,c=0,1;\beta=1,...,\lambda}$,

$\{A(\lambda)^{b}_{[k]0}, C(\lambda)^{\beta b}_{[k]0}\}_{k=m+1,...,m+n;b=0,1;\beta=1,...,\lambda}$,

$\{A\dagger(\lambda)^{b}_{[k]0}\}_{k=1,...,l;b=0,1}$,

$\{A\ddagger(\lambda)^{\alpha bc}_{[k]000}, C\ddagger(\lambda)^{\alpha\beta bc}_{[k]000}\}_{k=1,...,m;b,c=0,1;\beta=1,...,\lambda}$,

$\{A\ddagger(\lambda)^{\alpha b}_{[k]0}, C\ddagger(\lambda)^{\alpha\beta b}_{[k]0}\}_{k=m+1,...,m+n;b=0,1;\beta=1,...,\lambda}$,

$\{A\dagger\ddagger(\lambda)^{\alpha b}_{[k]0}\}_{k=1,...,l;b=0,1}$,

$\{Z'(\alpha)\}$,

$\{A'(\lambda)^{\alpha bc}_{[k]000}, C'(\lambda)^{\alpha\beta bc}_{[k]000}\}_{k=1,...,m;b,c=0,1;\beta=1,...,\lambda}$,

$\{A'(\lambda)^{\alpha b}_{[k]0}, C'(\lambda)^{\alpha\beta b}_{[k]0}\}_{k=m+1,...,m+n;b=0,1;\beta=1,...,\lambda}$,

$\{A\dagger'(\lambda)^{\alpha b}_{[k]0}\}_{k=1,...,l;b=0,1}, \ldots]$

Then, $\theta = \mathrm{Hash}(E,G,S) \bmod q$ is calculated to obtain the following expression.

$$[z''(\alpha)]G = Z'(\alpha) + [\theta]Z(\alpha)$$

After that, the following expressions are checked.

With respect to all $k=1,\ldots,m,\ b,c \in \{0,1\},\ \beta=1,\ldots,\lambda$.

$$[z''(\alpha)]\,A(\lambda)bc_{[k]000} = A'(\lambda)\alpha bc_{[k]000} + [\theta]\,A\sharp(\lambda)\alpha bc_{[k]000}$$

$$[z''(\alpha)]C(\lambda)\beta bc_{[k]000} = C'(\lambda)\alpha\beta bc_{[k]000} + [\theta]C\sharp(\lambda)\alpha\beta bc_{[k]000}$$

With respect to all $k=m+1,\ldots,m+n,\ b \in \{0,1\},\ \beta=1,\ldots,\lambda$.

$$[z''(\alpha)]A(\lambda)b_{[k]0} = A'(\lambda)\alpha b_{[k]0} + [\theta]A\sharp(\lambda)\alpha b_{[k]0}$$

$$[z''(\alpha)]C(\lambda)\beta b_{[k]0} = C'(\lambda)\alpha\beta b_{[k]0} + [\theta]C\sharp(\lambda)\alpha\beta b_{[k]0}$$

With respect to all $k=1,\ldots,l,\ b \in \{0,1\}$.

$$[z''(\alpha)]A\dagger(\lambda)b_{[k]0} = A\dagger'(\lambda)\alpha b_{[k]0} + [\theta]A\dagger\sharp(\lambda)\alpha b_{[k]0}$$

[Separate description G]

The calculator $_U(\alpha)$ uniquely and randomly generates

$z'(\alpha) \in_R Z/qZ$ and generates $A\dagger'_{[k]} = [z'(\alpha)]A\dagger b\lceil k\rfloor_{[k]}$ with respect to k=1,...,l. The certifier generates the following data.

$$S = [\{A^{\dagger b^{\lceil k \rfloor}}_{[k]}\}_{k=1,\ldots,l},$$

$$\{A^{\dagger\ddagger}_{[k]}\}_{k=1,\ldots,l},$$

$$\{A^{\dagger'}_{[k]}\}_{k=1,\ldots,l}]$$

Then, the certifier generates the following data.

$$\theta = \mathrm{Hash}(E, G, S) \bmod q$$

The certifier generates the following data.

$$z''(\alpha) = \theta\, z(\alpha) + z'(\alpha) \bmod q$$

The certifier sets the following data as the proof.

$$\{A^{\dagger\ddagger}_{[k]}\}_{k=1,\ldots,l}, z''(\alpha)$$

The verification method for the above-mentioned proof is as follows. The certifier generates the following data.

$$S = [\{A^{\dagger b^{\lceil k \rfloor}}_{[k]}\}_{k=1,\ldots,l},$$

$$\{A^{\dagger\ddagger}_{[k]}\}_{k=1,\ldots,l},$$

$$\{A^{\dagger'}_{[k]}\}_{k=1,\ldots,l}]$$

The certifier generates the following data.

$$\theta = \mathrm{Hash}(E, G, S) \bmod q$$

When the certifier confirms the following data, the certifier receives the

proof.

$$[z''(\alpha)]A^{\dagger b\lceil k\rfloor}{}_{[k]} = A^{\dagger}{}_{[k]} + [\theta]\, A^{\dagger\ddagger}{}_{[k]}$$

Second Embodiment

A second embodiment of the present invention will be described with reference to Figs. 14 to 18.

In this embodiment, as shown in Fig. 14, each of N calculation devices 1401 has a calculation device 1403. Hereinafter, these calculation devices are referred to as U1,...,UN in order. For the sake of notation, UN is also referred to as U0.

[Method outline]

[Data flow]

A data flow according to the second embodiment will be described with reference to Fig. 16.

First of all, U1 of the calculation device 1401 calculates DATA00. This is referred to as "a calculation of the zero-th cycle" (1701).

Next, "a calculation of the first cycle" is performed.

U1 calculates $DATA_1{}^1$ from $DATA_0{}^0$ to transmit DATA11 to U2 (1711).

Next, U2 calculates $DATA_1{}^2$ from DATA11 to transmit DATA12 to U3 (1712).

Hereinafter, the data is transmitted in turn. UN calculates $DATA_1{}^N$ from $DATA1^{N-1}$ to transmit DATA1N to U1 (1710). At this point, the calculation of the first cycle is ended.

Next, "a calculation of the second cycle" is performed.

U1 calculates $DATA_2{}^1$ from DATA1N to transmit DATA21 to U2 (1721).

Next, U2 calculates $DATA_2{}^2$ from DATA21 to transmit DATA21 to U3 (1722).

Hereinafter, the data is transmitted in turn. UN calculates $DATA_2{}^N$ from $DATA_2{}^{N-1}$ to transmit DATA2N to U1 (1720). At this point, the calculation of the

second cycle is ended.

Next, "a calculation of the third cycle" is performed.

U1 calculates $DATA_3^1$ from DATA2N to transmit DATA21 to U2 (1731).

Hereinafter, the data is transmitted in turn. The protocol is completed when UN completes the calculation of DATA3N from $DATA_3^{N-1}$.

[Input and output of the respective calculation devices 1401]

Next, input and output of data exchanged among the respective calculation devices 1401 will be described with reference to Fig. 14.

Circuit information 1404 and a circuit partial input 1402 are input to the respective calculation devices 1401.

Herein, a case will be described in which the number of fan-ins of elements other than input elements represented by the circuit information 1404 is 2.

An input bw to an input element w of the circuit is secretly owned by one of U1, ..., UN. U1 is also referred to as UN+1.

A combination of input bits secretly owned by UI corresponds to the circuit partial input 1402.

A circuit in which input elements i of the circuit represented by the circuit information 1404 are regarded as elements for outputting bw when any input is received is hereinafter expressed as C[1].

The number of fan-in of all gates of C[1] is 2.

Fan-ins at the upper left and the lower right are referred to as L(w) and R(w).

When the circuit information 1404 and the circuit partial input 1402 are input to the respective calculation devices 1401, U1 firstly performs the calculation of the zero-th cycle while following a procedure to be described later.

The input and output in the calculations of the first to third cycles have the same data configuration. Data to be transmitted to UI+1 in the calculation of an i-

th cycle by the user with respect to i=1,2,3 is hereinafter expressed as DATAil. DATAi-1N is also referred to as DATAi0.

The result of the calculation performed by U1 in the calculation of the zero-th cycle is referred to as DATA01.

$DATA_i^l$ has a format of $DATA_i^l = DATA_i^{l-1} || BODY_i^l || PROOF_i^l || SIG_i^l$.

DATAil-1 is a message sent from Ul-1. BODYil is a main body of the message. PROOFil is a validity proof text of BODYil. SIGil is a signature of Ul for DATAil-1 || BODYil ||PROOFil.

The outline of the calculation of the first to third cycles will be described.

In the calculation of the i-th cycle, first of all, Ul receives DATAil-1 from Ul-1 (1501).

(Only U1 of the first cycle exceptionally uses DATA01 made by itself).

When receiving $DATA_i^{l-1}$, Ul verifies all validity proof texts PROOF11, ...,PROOFl-11 (1502). Next, Ul verifies all signature text SIG_i^1, ..., SIGl-11 (1503).

Only in the case of the first cycle and also l=1, the calculation of 1504 is performed. Next, Ul performs random number generation (1505). Then, Ul uses the random number to perform the main calculation to generate BODYil (1506). After the main calculation is completed, Ul generates a validity proof text PROOFil of BODYil (1507). Then, Ul generates a signature text SIGil with respect to DATAil-1 || BODYil || PROOFil (1508).

Finally, Ul transmits DATAil =DATAil-1 || BODYil || PROOFil ||SIGil to Ul+1 (1509).

[Symbol]

Hereinafter, symbols used in this specification will be described.

[Encryption method E[27]]

G[1] denotes an abelian group with a difficult DDH problem (for example, an ellipse curve group on an infinite distance on an ellipse curve group on a finite

field), p denotes an order of G[1], and O denotes a zero element of G[1].

η is treated as a symbol and the following symbols are defined. It should be noted that with respect to $P[|0|] \in G[1]$, (P[|0|],0) is abbreviated to be simply expressed as P[|0|] and regarded as the following.

$$F_p \subset B[12],\ G[1] \subset G[12B]$$

Then, a sum on G[12B] is defined by a sum for each component to obtain the following.

$W[12] = B12^{\kappa}$ is set, establishing

$G[12W] = G[12B]^{\kappa}$

$w[2] \in W[12]$

α component of w[2]..... is expressed as [2|α|]. A sum and a product on W[12] are defined by a sum and a product for each component. Then, a sum and a scalar multiple on G[12W] are defined by a sum and a scalar multiple for each component.

- $B[12]=F_p[\eta]/(\eta^2-1)$,
- $\phi[24](1)=1,\ \phi[24](0)=\eta$,
- $G[12B]=G[1]^2$,
- $aP[2]=(a[|0|]P[|0|]+a[|1|]P[|1|], aa[|0|]Pa[|1|]+a[|1|]Pa[|0|])$
- $W[12]=B[12]^{\kappa}$,
- $e[|i|]=(0,\cdots,0,1,0,\cdots,0)$ (only i-th is 1)
- $\phi[2] : F_p \rightarrow W[12]$ is set as $x \rightarrow \Sigma_{\alpha}\ \phi[2](x[|\alpha|])e[|\alpha|]$
- ...
- Wherein
  - $P[2]=(P[|0|],P[|1|])$, $P[23]=(P[3|0|],P[3|1|]) \in G[12B]$, $a=a[|0|]+a[|1|]\eta \in B[12]$, $P[2]=(P[|0|],P[|1|]) \in G[12B]$,
  - κ : the number of bits of p,
  - $x=x[|k-1|] \,||\, \cdots \,||\, x[|0|]$,
  - $G[12W]=G[12B]^{\kappa}$,

The encryption method E[27] is a resemblance of an ellipse ElGamal

encryption in G[12W].

A secret key space is denoted by Fp, a public key space is denoted by G[12W]2, a plaintext space is denoted by G[12W], and a random number space is denoted by W[12].

In order to generate a key, a key satisfying P[|0|],P[|1|]≠O if P=P[|0|]+ηP[|1|] is arbitrarily selected. By randomly selecting $a \in F_p$, Q=aP is satisfied. a denotes a secret key and (P,Q) denotes a public key. In order to cipher a plain text M, $r \in B[12]$ is uniquely and randomly selected to calculate a cipher text (P[3],Q[3])=(rP,M+rQ). In order to decipher (P[3],Q[3]), Q[3]-aP[3] may be calculated.

[Encryption methods E[2],E[25]]

The following data is defined:

- K[1]={ {x[|wWh|]} (w∈C[1],W∈{L,R},h∈{0,1} are run through) | x[|wWh|]∈F
- A[12]={ { a[2|wWijk|]} ({w∈C[1],W∈{L,R},i,j,k∈{0,1}} are run through)|a[
- A[125]={ { A[25|wWijk|i[6]j[6]k[6]]({w∈C[1],W∈{L,R},i,i[6],j,j[
  |A[25|wWijk|i[6]j[6]k[6]]∈W[12]}
- G[1|K|]={ {P[|wWh|]} ({w∈C[1],W∈{L,R},h∈{0,1}} are run through)|P[|wWh
- G[12|A|]={{ P[2|wWijk|]} (w∈C[1],W∈{L,R},i,j,k∈{0,1}})|P[2|wWi.
- G[124|A|]={ { P[24|wWijk|i[6]j[6]k[6]]}
  ({w∈C[1],W∈{L,R},i,j,k,i[6],j[6],k[6]∈{0,1}} are run through)
  |P[24|wWijk|i[6]j[6]k[6]]∈G[12W].
- aA=(aA[|1|],aA[|2|])
- a[5]A[5]=(a[5]A[5|1|],a[5]A[5|2|])

Wherein

- a ∈A[12]
- A=(a[|1|],A[|2|]) ∈ G[12|A|]^2
- a[5] ∈ A[125]
- A[5]=(A[5|1|],A[5|2|]) ∈ G[124|A|]^2

A wWh component of the original x of κ is referred to as x[|wWh|]. A wWijk component of the original x[2] of A[12] is referred to as x[|wWijk|]. A component of the original A[25] of A[125] is referred to as x[|wWijk|i[6]j[6]k[6]].

A sum, a product, and a scalar multiple of a multiple sequence alignment are defined by a sum and a product for each component. However, exceptionally, only a product of the originals of A[125] and a scalar multiple of the original of A[125] of the original of G[124|A|] are defined as follows.

- a[25]*b[25]=Σ i[7], j[7], k[7]
  a[25|wWijk|i[7]j[7]k[7]]
  b[25|wWi[7]j[7]k[7]|i[6]j[6]k[6]],
- a[25]*P[24]=Σ a[25|wWijk|i[7]j[7]k[7]]
  P[24|wWi[7] j[7] k[7]|i[7]j[7]k[7]]

  (A sum related to i[7], j[7], k[7])

The following symbols are defined.

- E[25][(Z[2]|s[25])](M[25])
  ≡ {E[27][(Z[2]|s[25|wWijk|i[6]j[6]k[6]])](M[25|wWijk|i[6]j[6]k[6]])}
- E[25][(Y[2]|s[25])](M[25])
  ≡ E[25][(s[25]|Z[25])](M[25])
- E[2][(x[2]|r[2])](M[2])
  ≡ {E[27][(x[2|wWijk|]|r[2|wWijk|])](M[2|wWijk|])
- E[2][(Y[2]|r[2])](M[2])
  ≡ E[2][(x[2]|r[2])](M[2])

• Wherein

- M[25]={M[25|wWijk|i[6]j[6]k[6]]},
  M[25|wWijk|i[6]j[6]k[6]] ∈ G[12W]
- Z[2]={Z[25|wWijk|i[6]j[6]k[6]]},
  (Z[25|wWijk|i[6]j[6]k[6]] ∈ G[12W]2)
- s[25]={s[25|wWijk|i[6]j[6]k[6]]} ∈A[125]
- Y[2]=(P[2], r[2])∈G[12W],
- Z[25]={Y[2]}[|wWijk|i[6]j[6]k[6]]
- M[2]={M[2|wWijk|]}[|wWijk|] (M[2|wWijk|] ∈ G[12W])
- x[2]=(P[2], Q[2])=({P[2|wWijk|]}, { Q[2]
  [|wWijk|]})
  (P[2|wWijk|], Q[2|wWijk|]∈G[12W]),
- r[2]={r[2|wWijk|]}, r[2|wWijk|]∈ {}W[12]

The encryption methods E[2], E[25] are referred to as a multiple sequence alignment ElGamal encryption on a true value group ring and an extended multiple sequence alignment ElGamal encryption on the true value group ring, respectively.

[Other symbol]

- h[:1|w|](ijk))=(i□[|w|]j)○k
- F[25:1| λ[|1|], λ[|2|], λ[|2|]|](x[2])=
  {x[2:1|wh[|w|](ijk)|]
  δ(i,i[6]○λ[|1w|]),
  δ(j,j[6]○λ[|2w|]),
  δ(k,k[6]○λ[|3w|])}

(A field related to wWijk[6]jj[6]kk[6])

- J[2](x[2]) ∈ A[12]=J[2](x[2])={x[2|W(w)i[W]|]、
- π[2](a[25])={Σa[25|wWijk|i[6]j[6]k[6]]}[|ijk|]}

(The sum is a sum related to i[6]、j[6]、k[6])

Wherein the following is defined.

- λ[|1|]={λ[|1w|]}、 λ[|2|]={λ[|2w|]}、
  λ[|3|]={λ[|3w|]} : A field of bit (A field related to w∈C[1])
- x[2]={x[2|wWh|]} x[2|wWh|]∈W[12],
- 「○」 :An exclusive OR for each bit
- 「□[|w|]」 :An operator for a calculation with an element w

(It should be noted that when w is the input element, b_w is owned by one of i□[|w|]j=1(if b_w is U_1, ···, $U_{|-1}$,
i□[|w|]j=0 (othewise)

- δ(i,i[6]) : Kronecker delta
- i[W]=i (if w=L), i[W]=j (if w=R),
- a[25]={a[25|wWijk|i[6]j[6]k[6]]} ∈ A[125]

From a simple calculation, it is understood that the following is established:

- (a[25]*b[25])c[25]=a[25]*(b[25]*c[25]),
- (a[25]*b[25])P[2]=a[25]*(b[25]{}P[2]),
- F[25:1|λ[3|1|], λ[3|2|], λ[3|3|]|](1)
  *F[25:1|λ[|1|], λ[|2|], λ[|3|]|](x[2])
  ≡
  F[25:1|λ[3|1|]○λ[|1|], λ[3|2|]○λ[|2|], λ[3|3|]○λ[|3|]|](x[2])
- F[25:1|λ[|1|], λ[|2|], λ[|3|]|](x[2])
  *F[25:1|0,0,0|](x[23])=F[25|λ[|1|], λ[|2|], λ[|3|]|](x[2]○x[23])

  Wherein

  - a[25],b[25],c[25]∈A[125],
  - a[2],b[2]∈A[12],
  - x[2],x[23] ∈ κ,
  - λ[|1|]={λ[|1w|]}, λ[|2|]={λ[|2w|]}
    λ[|3|]={λ[|3w|]},
    λ[3|1|]={λ[|1w|]},
    λ[3|2|]={λ[|2w|]},
    λ[3|3|]={λ[|3w|]} : Bit array
  - l ∈ {1,…,N}

[Detail of the calculation of the zero-th cycle]

A detail of the calculation of the zero-th cycle will be described.

A description will be given of a method in which U1 calculates BODY10.

U1 calculates the following expressions.

- x[:0|wWh|]=0,
- x[2:0|wWh|]
  $\overline{=}$
  ϕ[2](x[:0|wWh|]),
- x[:0]
  $\overline{=}$
  {x[:0|wWh|]},
- x[2:0]
  $\overline{=}$
  {x[2:0|wWh|]},
- λ[:0]={λ[:0|w|]={0},
  λ[L:0]={λ[:0|L(w)|]={0},
  λ[R:0]={λ[:0|R(w)|]={0},
- r[2:0]={r[2:0|wWijk|]}={0} (A field related to wWijk)
- y[:0]=0,
  r[2:0]=y[:0]P[2](=O), Y[2:0]=(P[2],r[2:0])
- s[2:0|S|]={(s[2:0|S|])[|wWijk|i[6]j[6]k[6]]}
  ={0}[|wWijk|i[6]j[6]k[6]],
- s[2:0|T|]={(s[2:0|T|])[|wWijk|i[6]j[6]k[6]]}
  ={0}[|wWijk|i[6]j[6]k[6]],
- s[25U:0]={(s[25U:0|wWijk|i[6]j[6]k[6]]}
  ={0}[|wWijk|i[6]j[6]k[6]],
- (S[2:0],T[2:0],U[25:0]) = (E[2][(Y[2:0]|s[2:0|S|])](r[2:0]P[2])
- E[2][(Y[2:0]|s[2:0|T|])](r[2:0]Q[2:0]),
- E[25][(Y[2:0]|s[25R:0])](F[25:||λ[L:0],λ[R:0],λ[:0]|](x[2:0])P[2])
- $BODY_1^0$ =
  (Q[2:0],r[2:0]) || (s[2:0],T[2:0],U[25:0]),
  Wherein
  - w∈{}C[1],W∈ {L,R},h∈{0,1},i,j,k∈{0,1}

[Detail of calculation of the first cycle]

A multi party calculation of the first cycle will be described. The description will be given with reference to Fig. 15.

[Detail of data obtainment 1501]

Each U1 receives data DATA1l-1 from Ul-1. (Only U1 calculates data DATA1l-1=DATA10 by its own exceptionally).

[Detail of validity 1502 of the proof text in data]

When DATA1l-1 || BODY1l-1 || PROOF1l-1 || SIG1l-1 || is sent from Ul-1, Ul checks the validity of $PROOF_1^1$, ...,$PROOF_1^{l-1}$. A detail of this validity check

will be described later.

[Detail of validity 1503 of the signature text in data]

UI checks the validity of SIG11, ..., SIG1I-1. Furthermore, a hash value P of RAND is calculated to check P[2]=(e[|0|]+...+e[|κ-1|])(1+η)P.

[Detail of calculation performed only by U1]

U1 first select a random number RAND to set a hash value of RAND as follows.

$$P \in G[1]$$

Then, P[2]=(e[|0|]+...+e[|κ-1|])P[2B] is set to establish BODY1-1= RAND || P[2]. After that, BODY10 is generated by itself while following a procedure to be described later and further PROOF10=SIG10=ε is set, thereby establishing DATA10=RAND||BODY10||PROOF10||SIG10.

[Detail of random number generation]

U1 randomly selects the following (but the selection is made while descriptions in brackets are satisfied):

- {x[#I|wh|]} (x[#I|w[t]0|]}=0,
  x[0#I|w[t]0|]=1,
  x[#I|wh|]∈K[I], The most significant bit of each x[#I|wh|] is 1).
- {x[|wWh|]},x[|wWh|] ∈{0,1} (x[#I|wLh|]○x[#I|wRh|]=x[#I|wh|])
{w∈C[1],W∈{L,R},h∈{0,1}}
- r[2]={r[2|wWijk|]}∈A[12],
- r[2|wWijk|]∈F_p^{κ}⊂B[12]$^{\kappa}$=W[12],
- s[2|S|]={(s[2|S|])[|wWijk|]}∈A[12],
- s[2|T|]={(s[2|T|])[|wWijk|]}∈A[12],
- s[25U]={s[25U|wWijk|i[6]j[6]k[6]]}∈A[125],
- λ[#I]={λ[#I|w|]} (Field of the bit, λ[|w[t]|]=0)
- y[#I]∈ F_p⊂ κ

•Wt: output element

With respect to the input element w, x[#l|L(w)0|]=x[#l|R(w)1|]=0 is defined for form's sake.

The following symbols are defined.

- x[:l|wWh|]=⊕x[#γ|wWh|],
  (An exclusive OR in the range of γ≦l is taken).
  x[:l]={x[:l|wWh|]},
  x[2#l|wWh|]=φ[2](x[#l|wWh|]),
  x[2#l]={x[2#l|wWh|]},
  x[2:l|wWh|]=φ[2:l](x[|wWh|]),
  x[2:l]={x[2:l|wWh|]},
- λ[#l]={λ[#l|w|]},
  λ[L#l]={λ[#l|L(w)|]},
  λ[R#l]={λ[#l|R(w)|]},
  λ[:l|w|]=Σλ[#γ|w|], (A sum in the range of γ≦l is taken)
  λ[:l]={λ[:l|w|]},
  λ[L:l]={λ[:l|L(w)|]},
  λ[R:l]={λ[:l|R(w)|]},
- Y[2:l]=(P[2],y[:l]P[2]),
  y[:l]=Σy[#l], (A sum in the range of γ≦l is taken)

The following fact regarding the proposed method can be mentioned. When the users comply with the protocol, the data BODY1l sent by each Ul satisfies the following.

- BODY1l = (Q[2:l],r[2:l]) || (s[2:l],T[2:l],U[25:l]),
- (Q[2:l],r[2:l])=J[2](x[2:l])P[2],y[:l]P[2]),
- (s[2:l],T[2:l],U[25:l])=(E[2][(Y[2:l]|s[2][S:l])](r[2:l]P[2]),
E[2][(Y[2:l]|s[2:l|T|])](r[2:l]Q[2:l]),
E[25][(Y[2:l]|s[25U:l])](F[25:l|λ[L:l],λ[R:l],λ[:l]|](x[2:l])P[2])

Wherein

- s[2:I|S|]
  $\sum_{\gamma \leq I}$ s[2#γ|S|] (A sum in the range of $\gamma \leq I$ is taken) ;
- s[2:I|T|]
  $\sum_{\gamma \leq I}$ s[2#γ|T|],
-
  s[25U:I]= [235U:I]+s[25U#I],
  s[235U:I]
  $\sum_{\gamma \leq I}$ F[25:I|λ[L#I], λ[R#I], λ[#I]|](1)
  * s[25U:I-1]
  * F[25:I|0,0,0|](x[2#I]),

[Detail of mail calculation 1506]

A mail calculation 1506 of the multi party calculation of the first cycle will be described with reference to Fig. 17.

A description will be given of a method in which UI calculates BODY1I.

[Detail of mail calculation 1701]

First of all, UI calculates the following expressions.

- Q[2:I]=J[2]x[2#I])Q[2:I-1]
- r[2:I]=y[#I]P[2]+r[2:I-1]

[Data exchange 1702]

Then, the following expressions are calculated.

- S[23:I]=r[2#I]s[2:I-1]
- T[23:I]=r[2#I]J[2](x[2#I])T[2:I-1]
- U[235:I]=F[25:I|λ[L#I],λ[R#I],λ[#I]|](1)*U[25:I-1]*F[25:I|0,0,0|](x[2#I])

It should be noted that the following can be satisfied:

- Q[2:I]=J[2](x[2:I])P[2],
- r[2:I]=y[:I]P[2],
- S[23:I]=E[2][(Y[2:I-1]|s[2:I-1|S|])](r[2:I]P[2]),
- T[23:I]= E[2][(Y[2:I-1]|s[2:I-1|T|])](r[2:I]Q[2:I]),
- U[235:I]=E[25][(Y[2:I-1]|s[235U:I])](F[25:I|λ[L:I],λ[R:I],λ[:I]|](x[2:I])P[2]).

Wherein

•s[235U:I]=F[25:I|λ[L#I],λ[R#I],λ[#I]|](1)

*s[25U:I-1]*F[25:I|0,0,0[](x[2#I])

[Reencryption-secret key conversion 1703, reencryption-random number conversion 1704, validity proof text generation 1507, signature text generation 1508, data transmission 1509]

Finally, UI calculates the following expressions. (Reference numerals in figures are in parentheses).

•(1703)s[233:I]=S[23:I|1|],S[23:I|2|]+y[#I]s[2:I|1|]),

·T[233:I]=T[23:I|1[],T[23:I|2|]+y[#I]T[2:I|1|]),

·T[2335:I]=(U[235:I|1|],U[235:I|2|]+y[#I]U[25:I|1|]),

•(1704)

s[2:I]=s[233:I]+E[2][(Y[2:I]|s[2#1|S|])](O),

T[2:I]=T[233:I]+E[2][(Y[2:I]|s[2#1|T|])](O)

U[235:I]=(U[235:I|1|],U[235:I|2|]),

U[25:I]=T[2335:I]+E[25][(Y[2:I]|s[25U#I])](O),

(s[2:I],T[2:I],U[25:I])

=(E[2][(Y[2:I]|s[2:I|S|])](r[2:I]P[2]),

E[2][(Y[2:I]|s[2:I|T|])](r[2:I]Q[2:I]),

E[25][(Y[2:I]|s[25U:I])]

(F[25:I][(λ[L:I],λ[R:I],λ[:I]|)](x[2:I]P[2]),

•(1507)

•(1508)

·A signature of BODY1I=s[2:I]·DATA1I-1 || BODY1I-1 || PROOF1I-1

SIG1I is generated.

•(1509)DATA1I=DATA1I-1 || BODY1I || PROOF1I || SIG1IDATA1I is set as UI+1

Wherein

• S[23:I]=E[2][(Y[2:I-1]|s[2:I-1|S|])](r[2:I]P[2]),

- T[23:I]=E[2][(Y[2:I-1]|s[2:I-1|T|])](r[2:I]P[2]),
- U[235:I]=E[25][(Y[2:I-1]|s[25U:I-1])](r[25:I]P[2]),
- S[23:I]=(S[23:I|1|],S[23:I|2|]),
- T[23:I]=(T[23:I|1|],T[23:I|2|]),
- T[23:I]=(T[23:I|1|],T[23:I|2|]),

A detail of the validity proof text is long so the description will be given later.

DATA1I=(Q[2:I],r[2:I]) || (s[2:I],T[2:I],U[25:I]) is sent to UI+1.

It should be noted that the following fact can be mentioned:

- S[233:I] =E[2][(Y[2:I]|s[2:I-1|S|])](r[2:I]P[2]),
- s[2:I] = E[2][(Y[2:I]|s[2:I|S|])](r[2:I]P[2]).

[Detail of calculation of the second cycle]

A detail of the calculation of the second cycle will be described with reference to Fig. 15.

[Detail of the data obtainment 1501]

Each UI receives data DATA2I-1=DATA2I-2 || BODY2I-1 || PROOF2I-1 || SIG2I-1 from UI-1.

[Verification 1502 of proof text in data, Verification 1503 of signature text in data, calculation 1504 performed only by U1, detail of random number generation 1505, cipher calculation 1801 on Fp]

Calculations described below are performed.

- (1502) Validity verification of PROOF21 || ... || PROOF2I-1 (the detail will be described below)
- (1503) Validity verification of SIG21 || ... || SIG2I-1
- (1504) No process is performed.
- (1505) No process is performed.

[Detail of the main calculation 1506]

A detail of the main calculation in the calculation of the second cycle will be described with reference to Fig. 18.

[Calculation of ciphers on Fp 1801]

Hereinafter, this section omits the additional character 「:N」.

UI performs the following calculations:

•s[2U]=π[2](s[25U]),

•m[2S]= r[2],

•m[2T]= r[2]J[2](x[2]),

•m[2U]= F[25|λ[L],λ[R],λ|](x[2]),

•U[2]=π[2](U[25])

•(s[2],T[2],U[2])

=E[2][(Y[2]|s[2|S|])](m[2S]P[2]),

E[2][(Y[2]|s[2|T|])](m[2T]P[2]),

E[2][(Y[2]|s[2U])](m[2U]P[2])

Wherein

•X={(P,x[|wh|]P)}, Y={(P,yP)},

•E[1][(r|X)](A)={E[7][(r[|wWijk|]|X)](A[|wWijk|]P)}

•E[7]: An encryption function of an ellipse ElGamal encryption method.

Then, the following calculations are performed.

- (s[24V|wWijk|]=Σs[2V|αwWijk|],(A sum related to α is taken)
- (m[24V|wWijk|]=Σ2αm[2V])[|αwWijk|],(A sum related to α is taken)
- (s[24V|wWijk|]P[2B||]=Σ2αs[2V|αwWijk|]P[2]),(A sum related to α is taken)
- s[24V]P[2B]=(s[4V|0,wWijk|]P,s[4V|1,wWijk|]P),
- (m[24V|wWijk|]+(s[24V|wWijk|]y)P[2B]

  =Σ2αm[2V|αwWijk|]+s[2V|αwWijk|])yP[2B](A sum related to α is taken)
- .
- .

• Wherein

  - (s[2V|wWijk|]

  =Σs[2V|αwWijk|]e[α] (A sum related to α is taken)
- (s[24V|wWijk|]=((s[V|0,wWijk|],(s[V])[|1,wWijk|])
- (m[2V|wWijk|]=m[2V|wWijk|]

  =Σm[2Vα|wWijk|]e[|α|],(A sum related to α is taken)
- m[24V|wWijk|]=(m[4V|0,wWijk|],m[4V|1,wWijk|]),
- P[2]

  =Σe[|α|]P[2B] (A sum related to α is taken)
- P[2B]=(P,P), P∈{}G[1],
- V[2]

  =E[2][(Y[2]|s[2V])](r[2]P[2])

  =(s[2V]P[2],m[2V]P[2]+s[2V]r[2]),
- s[2V|wWijk|]P[2]

  =Σ(s[2Vα|wWijk|]e[|α|]P[2B], (A sum related to α is taken)
- (m[2V|wWijk|]+s[2V|wWijk|]・y)P[2B] ∈ W[12]

  =Σ(((m[2V])[|α|])[|wWijk|]+s[2V|α||wWijk|]yP[2B]e[|α|]。
  (A sum related to α is taken)
- (m[24V|wWijk|]+s[24V|wWijk|]y)P[2B]

  =(m[4V|0,wWijk|]+s[V|0,wWijk|]y)P,
  (m[4V|1,wWijk|]+s[V|0,wWijk|]y)P

Thus, UI can obtain the following:

•s[V|0,wWijk|]P, s[V|1,wWijk|]P,

•m[4V|0,wWijk|]+s[V|0,wWijk|]y)P,

m[4V|1,wWijk|]+s[V|0,wWijk|]y)P,

Moreover, the following calculations are performed.

•R=yP, Y=(P,R)

•(s[V|0,wWijk|]P,

(m[4V|0,wWijk|]+s[V|0,wWijk|]y)P

=E[(s[V|0,wWijk|]|Y)](m[4V|0,wWijk|]P),

•(S,T+U)

=(E[3][(s[4S0],s[4T0]+s[4U0]|Y)],E[1][(r[40]|J(X))](F[|λ[L],λ[R],λ|](x))

•{(Θ[|wWijk|],Θ'[|wWijk|])}

=E[3][(s[4S0],s[4T0]+s[4U0]|Y)],E[1][(r[40]|J(X))](F[|λ[L],λ[R],λ|](x))

•Θ[&0]={(Θ[|wWij0|],Θ'[|wWij0|])}

•R[&0]=R0

Wherein

•r[2]=R[|0|]+R[|1|]η

[Calculation of Θ[&1] 1802, detail of the validity verification text generation 1507, detail of the signature text generation 1508, detail of the data transmission 1509]

Furthermore, the following calculations are performed:

•(1802)

•Θ[&1]=Θ[&I-1]-(Θ[|0|][&I-1],Θ[|1|][&I-1]-y[&I]Θ[|0|][&I-1]),

•BODY2I=Θ[&I],

Wherein

•BODY2I-1=Θ[&I-1],

•Θ[&I-1]=(Θ[&I-1][0],Θ[&I-1|1|])

•(1508)

•(1803)

•BODY2I = s[2:I]

•SIG2I: A signature is given to DATA2I-1 || BODY2I || PROOF2I

Then, DATA2l=DATA2l-1 || BODY2l || PROOF2l || SIG2l is transmitted to Ul+1 (1509).

[Detail of the calculation of the third cycle]

A detail of the calculation of the third cycle will be described with reference to Fig. 15.

When $DATA_3^{l-1}$=DATA3l-2 || BODY3l || PROOF3l || SIG3l is sent from U_{l-1} (1501), firstly, the validity of $PROOF_3^1$, ..., PROOF3l-1, SIG3l, ..., SIG3l-1 is checked (1502,1503). (1504) and (1505) are not the calculations of the third cycle. BODY3l=ε is established (1506) and PROOF3l=ε is established (1507). Then, a signature SIG3l for DATA3l-1 || BODY3l || PROOF3l is generated (1508) and DATA3l=DATA3l-1 || BODY3l || PROOF3l || SIG3l is established to send DATA3l to Ul (1509).

[Method of obtaining C[1]({b[|w|]}) 1405]

A method of obtaining C[1]({b[|w[]}) will be described.

First, as x[#l|L(w)0[]=x[#l|R(w)1[]=0 is established with respect to the input element w, {E[1][(|{X[|W(w)i[|W|]|]}})]x[|wWh[3](ij0)[])} is solved entirely to obtain Xx[|wWh[3](ij0)[]=x[|wWμ[w]|]. Then, x[3|w[]=x[|wμ[w]|]=x[|wLμ[w]|]∘{}x[|wRμ[w]|] is calculated to establish μ[w]=h[3](ij0).

Incidentally, with respect to each element w from the bottom to the stage of u-1, x[|μ[|w|]|] is found out.

The following calculations are performed to find out x[|μ[|w[]|] at the u stage.

•E[1][(|X[|L(w)μ[|L(w)|]|])]
(x[|{wWh[3](μ[|L(w)|]j0)}|]),
E[1][(|X[|R(w)μ[|R(w)|]|])]
(x[|wWh[3](iμ[|R(w)0)|]|]) (i,j=0,1)
are solved with use of x[|W(w)μ[|W(w)|]|].

•x[|wh[3]μ[|L(w)|]μ[|R(w)0[]|]=

∘x[|wWh[3]μL(w)μ[|R(w)0|]|]

(An exclusive OR with respect to W)

Wherein

•h[3](ijk)=h[|w|]((i∘λ[|L(w)|])(j∘λ[|R(w)|])(k∘λ[|w|]))

•μ[w]=b[|w|]∘λ[|w|] are set

•b[|w|]: The output of the element w

•x[3|w|]=x[|wμ[w]|]

x[|μ[|w[|t|]|]|]=μ[|[|w[|t|]|]|]=b[|[|w[|t|]|]|]=C[1]({b[|w|]}) is finally output.

[Validity proof for the calculation of the first cycle]

[Detail of the validity proof generation 1507 for the calculation of the first cycle]

The following symbols are defined.

- $A[12|F_p|]=\{\{a[2|wWijk|]\}|a[2|wWijk|]\in F_p\}$,
- $F[25:||a[|0|]b[|0|]c[|0|]|a[|1|]b[|1|]c[|1|]](u[2])\in A[12]$ is set as

$$F[25:||abc|a[|6|]b[|6|]c[|6|]](u[2])$$
$$=\overline{\{x[2|h[:||wa[|i\bigcirc i[6]|]b[|j\bigcirc j[6]|]c[|k\bigcirc k[6]|]|]|]|](ijk)\}}$$

- Wherein
  - w : element
  - $z[2]\in K[12]$,
  - $a[2]\in A[12]$,
  - $a[25]\in A[125]$,
  - $a[|0|],a[|1|],b[|0|],b[|1|],c[|0|],c[|1|],\in F_p$,
  - $u[2]\in K[12]$,

In this section, the additional character 「:|-1」 is omitted for the sake of simplicity.

The knowledge is proved through the following method.

First of all, a random number is put in a hash function to generate the following.

P[6]∈G[1]

The following calculations are performed.

•P[26B]=(1+η)P,

•P[26]=ΣP[26B]e[|α|], (A sum related to α is taken)

•NOT(λ)={NOT(λ[|w|])}

UI generates PROOF1I in a procedure described below. The following data is randomly and uniquely selected.

x[26#I]∈α,

ρ[26#I|S[23]|]∈A[12]2,

ρ[26#I|S[233]||2|]∈A[12],

ρ[26T[2&1]#I]∈A[12]2,

ρ[26T[233]#I]∈A[12]2,

ρ[26T[233]#I|2|]∈A[12],

ρ[246#I|T[233]||2|]∈A[12]2,

ρ[256#I|U[25&1]|]∈A[125]2,

ρ[256#I|λ|]∈A[125],

ρ[256#I|U[25&2]|]∈A[125],

ρ[256#I|U[25&3]|]∈A[125],

ρ[256#I|U[235:I]|]∈A[125]

ρ[256#I|T[2335:I]||2|]∈A[12].

Then, the following calculations are performed.

•C[2x[2]#I|[|αwWh|]|]=x[2#I|αwWh|]P[2B]+(x[26#I|αwWh|]P[26B],C[2x[2]#I]={Σe[|α

[]C[2x[2]#I[[|αwWh|]]|]}[|wWh|], (A sum related to α is taken)

•C[2#I|S[23]|] = S[23:I] +ρ[26S[23]#I]P[26],τ[S[23]#I]=ρ[26S[23]#I],

•C[2S[23]#I]=(C[2S[23]#1|1|],(C[2S[23]#1])[|2|]) is described as

(C[2S[23]#I|1|],C[2S[23]#I|2|]).

C[y,C[2S[23]]#I|1|]

=

yC[2S[23]#I|1|] +ρ[26S[233]#I|2|]P[26||]{A[12]},

C[2S[233]#I]=(C[2S[23]#I|1|],C[2S[233]#I|2|]),

C[2S[233]#I|2|]=C[2S[23]#I|2|]+C[2y,C[2S[23]]#I|1|],

τ[S[233]#I]=(τ[S[23]#I|1|],

τ[S[23]#I|2|]+yτ[S[23]#I|1|]+ρ[26S[233]#I]).

Wherein (τ[S[23]#I|1|], τ[S[23]#I|2|]) =(τ[S[23]#I]).

•C[2T[2&1]#I]=T[2&1:I]+ρ[26T[2&1]#I]P[26] is set as, τ[T[2&1]#I] = ρ[26T[2&1]#I],

•C[2T[233]#I]=T[233:I] +ρ[26T[233]#I]P[26],τ[T[233]#I]=ρ[26T[233]#I],

•C[2T[23:I]#I]=(C[2T[23:I]#I|1|],C[2T[23:I]#I|2|]) is described as

(C[2T[23:I]#I|1|],C[2T[23:I]#I|2|]).

C[y,C[2T[23:I]]#I|1|]=yC[2T[23:I]#I|1|] +ρ[26T[233]#I|2|]P[26A[12]],

C[2T[233]#I]=(C[2T[23:I]#I|1|],C[2T[233]#I|2|]),

C[2T[233]#I|2|]=C[2T[23:I]#I|2|]+C[2y,C[2T[23:I]]#I|1|],

τ[T[233]#I]=(τ[T[23:I]#I|1|],

τ[T[23:I]#I|2|]+yτ[T[23:I]#I|1|]+ρ[26T[233]#I]).

Wherein (τ[T[23:I]#I|1|], τ[T[23:I]#I|2|]) =τ[T[23:I]#I].

•K[2y,C[2T[233]]#I|1|]=y[4#I]C[2T[23:I]]+ρ[246T[233]#I|1|]P[26],

•c[25U[25&1]#I]=U[25&1:I]+ρ[256#I|U[25&1]|]P[26],τ[U[25&1]#I]=ρ[256#I|U[25&1]|].

•C[2λ#I]=λ[#I]P[2]+ρ[256#I|λ|]P,

•ρ[256NOT(λ)#I]=-ρ[256λ#I],C[2NOT(λ)#I]=P[2]-C[2λ#I],

•C[2U[25&2]#I]=F[25:I|λ[L]{}00|NOT(λ)[L]00](u[2])C[2U[25&1]#I]+ρ[256#I|U[25&2

]|]P[26],

Wherein NOT(λ)[L]={NOT(λ)[|L(w)|]}.

τ[U[25&2]#I]=F[25:I|λ[L]{}00|NOT(λ)[L]00](u[2])τ[U[25&1]#I]+ρ[256#I|U[25&2]|],

•C[2U[25&3]#I]=F[25:I|0λ[R]0|0NOT(λ)[R]0](u[2])C[2U[25&2]#I]+ρ[256U[25&3]#I]P[26],

Wherein

NOT(λ)[R]={NOT(λ)[R|R(w)|]}.

τ[U[25&3]#I]=F[25:I|0λ[R]0|0NOT(λ)[R]0](u[2])τ[U[25&2]#I]+ρ[256U[25&3]#I].

•C[2U[235:I]#I]=F[25:I|00λ|00NOT(λ)](u[2])C[2U[25&3]#I],

τ[U[235:I]#I]=F[25:I|00λ|00NOT(λ)](u[2])τ[U[25&3]#I],

•C[2U[235:I]#I]=((C[2U[235:I]#I])[|1|],(C[2U[235:I]#I])[|2|]) is described as (C[2U[235:I]#I|1|],C[2U[235:I]#I|2|]).

C[y,C[2U[235:I]]#I|1|]=yC[2U[235:I]#I|1|] +ρ[256#I|T[2335:I]||2|]P[26A[12]],

C[2T[2335:I]#I]=(C[2U[235:I]#I|1|],C[2T[2335:I]#I|2|]),

C[2T[2335:I]#I|2|]=C[2U[235:I]#I|2|] +C[2y,C[2U[235:I]]#I|1|],

τ[T[2335:I]#I]=(τ[U[235:I]#I|1|],

τ[U[235:I]#I|2|]+yτ[U[235:I]#I|1|]+ρ[256T[2335:I]#I]),

Wherein (τ[U[235:I]#I|1|], τ[U[235:I]#I|2|]) =τ[U[235:I]#I].

Furthermore, the following expressions are calculated.

●In the case of x[2|αwWh|]P[2B]=1

●With respect to each αwWh, x[246&1 | αwWh|]∈B[1] is randomly selected to calculate K[2&1 | αwWh|]= x[246&1 | αwWh|]P[26B]

● Randomly,

x[29& η | αwWh|],
c[2& η | αwWh|]∈B[1] is selected to calculate
K[2& η | αwWh|]
=
x[29& η | αwWh|]P[26]
−c[2& η | αwWh|](C[2x[2] | αwWh|]− ηP[2B])

●In the case of x[2|αwWh|]P[2B]=η.

●With respect to each αwWh, x[246& η | αwWh|]∈B[1] is randomly selected to calculate K[2& η | αwWh|]= x[246& η | αwWh|]P[26B]

● Randomly,

x[29&1 | αwWh|], c[2&1 | αwWh|]
∈B[1] is selected to calculate
K[2&1 | αwWh|]=x[29&1 | αwWh|]P[26]
−c[2&1 | αwWh|](C[2x[2]] | αwWh|]−P[2B])
and
(K[2&1 | αwWh|],
(K[2& η | αwWh|]

Moreover, the following data is uniquely and randomly selected.

x[24#I], x[246#I] ∈ A[12],

y[4#I]∈ F_p,

r[24#I]∈A[12F_p],
ρ[246S[23]#I]∈A[12]^2,

ρ[246S[233]#I|2|]∈A[12]^2,

s[24S#I]∈A[125], τ[4S[233]#I]∈A[125]^2,

r[24#I]∈A[12F_p],
ρ[246T[2&1]#I]∈A[12]^2,

ρ[246T[233]#I]∈A[12]^2,

s[24T#I]∈A[125], τ[4T[233]#I]∈A[125]^2,

ρ[2456U[25&1]#I]
∈ A[125]^2,

ρ[2456λ#I]∈A[125],
λ[4#I]∈F_p,

NOT(λ)[4#I],

=
{NOT(λ)[4#I|w|]},
NOT(λ)[#I|w|]
∈ F_p,
ρ[2456U[25&2]#I]∈ A[125],

ρ[2456T[2335:I]#I|2|]∈A[12]^2,

s[245U#I]∈A[125], τ[4T[2335:I]#I]∈A[125]^2,

Then, the following expressions are calculated.

- K[2x[2]#I]=x[24#I]P[2]+x[246#I]P[26],
- K[2J[2](x),Q[2]#I]=J[2](x[24])Q[2],

$$K[2y\#1]a=y[4\#1]P\Subset F_p\subset B[1],$$

•K[2r[2],s[2]#l]=r[24#l]s[2]+ρ[246S[23]#l]P[26],

•K[2y,C[2S[233]]#l|1|]=y[4#l]C[2S[23]]+ρ[246S[233]#l|1|]P[26],

•K[2s[24S]#l]=s[24S#l]Y[2]-τ[4S[233]#l]P[26],

•K[2r[2],T[2]#l]=r[24#l]T[2]+ρ[246T[2&1]#l]P[26],

•K[2J[2](x[2]),T[2&1]#l]=J[2](x[24#l])T[2&1]+ρ[246T[233]#l]P[26],

•K[2s[24T]#l]=s[24T#l]Y[2]-τ[4T[233]#l]P[26],

•K[25U[25&1]#l] =U[25:l-1]*F[25:l|0,0,0|](x[24#l])+ρ[2456U[25&1]#l]P[26],

•K[2λ,P[2]#l]=λ[4#l]P[2]+ρ[2456λ#l]P[26],K[2λ,C[2λ]#l]=λ[4#l]C[2λ#l]+ρ[2456λ#l]P[26],

•λ[4L#l]={λ[4#l|L(w)|]},NOT(λ)[4L#l]={NOT(λ)[4#l|L(w)|]},K[2U[25&2]#l]=F[25:l|λ[4L]00|NOT(λ)[4L]00](1)*U[25&1]+ρ[2456U[25&2]#l]P[26],

•λ[4R#l]={λ[4#l|R(w)|]},NOT(λ)[4R#l]={NOT(λ)[4R(w)#l]},K[2U[25&3]#l]=F[25:l|0λ[4R]0|0NOT(λ)[4R]0](1)*U[25&2]+ρ[2456U[25&3]#l]P[26],

•λ[4R#l]={λ[4R(w)#l]},NOT(λ)[4R#l]={NOT(λ)[4R(w)#l]},K[2U[235:l]#l]=F[25:l|00λ[4]|00NOT(λ)[4]](1)*U[25&3],

•K[2y,C[2T[2335:l]]#l|1|]=y[4#l]C[2U[235:l]]+ρ[2456T[2335:l]#l|1|]P[26],

•K[2s[245U]#l]=s[245U#l]Y[2]-τ[4T[2335:l]#l]P[26],

Furthermore, the following expressions are calculated.

c[#l]=Hash(DATA1l-1,BODY1l,C[2x[2]#l],C[2S[23]#l],(C[2S[23]#l|1|],C[2S[23]#l|2|]),C[y,C[2S[23]]#l|1|],C[2T[2&1]#l],C[2T[233]#l],(C[2T[23:l]#l|1|],C[2T[23:l]#l|2|]),C[2T[233]#l],K[2s[24T]#l]=s[24T#l]Y[2]c[25U[25&1]#l],C[2λ#l],C[2NOT(λ)#l],C[2U[25&2]#l],C[2U[25&3]#l],C[2U[235:l]#l],(C[2U[235:l]#l|1|],C[2U[235:l]#l|2|]),C[y,C[2U[235:l]]#l|1|],C[2T[2335:l]#l|2|],K[2&1|αwWh|],K[2&η|αwWh|],K[2x[2]#l],K[2J[2](x),Q[2]#l],K[2y#l],K[2r[2],s[2]#l],K[2y,C[2S[233]]#l|1|],K[2s[24S]#l],K[2r[2],T[2]#l],K[2J[2](x[2]),T[2&1]#l],K[25U[25&1]#l],K[2λ,P[2]#l],K[2λ,C[2λ]#l],K[2U[25&2]#l],K[2U[25&3]#l],K[2U[235:l]#l],K[2y,C[2T[2335:l]]#l|1|],K[2s[245U]#l],).

Wherein Hash denotes a Fp value hash function.

Then, the following expressions are performed.

•In the case of x[2|αwWh|]P[2B]=1.

•With respect to each αwWhc, [2&1|αwWh|]=c[#l]-c[2&η|αwWh|] is set to calculate x[29&1|αwWh|]=c[2&1|αwWh|](x[26|α|])[|wWh|]+x[246&1|αwWh|].

•In the case of x[2|αwWh|]P[2B]=η.

•With respect to each αwWh, c[2&η|αwWh|]=c[#l]-c[2&1|αwWh|] is set to calculate x[29&η|αwWh|]=c[2&η|αwWh|](x[26|α|])[|wWh|]+x[246&η|αwWh|], establishing (c[#l]c[2&1|αwWh|],x[29&1|αwWh|]),c[2&η|αwWh|],x[29&η|αwWh|])).

Moreover, the following expressions are performed.

•x[28#l]=c[#l]x[2#l]+x[24#l],x[268#l]=c[#l]x[26#l]+x[246#l],

•y[8#l]=c[#l]y[#l]+y[4#l],

•r[28#l]=c[#l]r[2#l]+r[24#l],ρ[268S[23]#l]=c[#l]ρ[26S[23]#l]+ρ[246S[23]#l],

•r[8#l]=c[#l]r[2#l]+r[24#l],ρ[268S[233]#l|2|]=c[#l]ρ[26#l|S[233]||2|]+ρ[246S[233]#l|2|],

•s[28S#l]=c[#l]s[24S#l]+s[24S#l], τ[8S[233]#l]=c[#l]τ[S[233]#l]+τ[4S[233]#l],

•r[28#l]=c[#l]r[2#l]+r[24#l],ρ[268T[2&1]#l]=c[#l]ρ[26T[2&1]#l]+ρ[246T[2&1]#l],

•ρ[268T[233]#l]=c[#l]ρ[26T[233]#l]+ρ[246T[233]#l],

•r[8#l]=c[#l]r[2#l]+r[24#l],ρ[268T[233]#l|2|]=c[#l]ρ[26T[233]#l|2|]+ρ[246#l|T[233]||2|],

•s[28T#l]=c[#l]s[24T#l]+s[24T#l], τ[8T[233]#l]=c[#l]τ[T[233]#l]+τ[4T[233]#l],

•ρ[2568U[25&1]#l]=c[#l]ρ[256U[25&1]#l]+ρ[2456U[25&1]#l],

•λ[8#l]=c[#l]λ[#l]+λ[4#l],ρ[2568λ#l]=c[#l]ρ[256λ#l]+ρ[2456λ#l],

•NOT(λ)[8#l]=c[#l]NOT(λ)[#l]+NOT(λ)[4#l],

ρ[2568U[25&2]#l]=c[#l]ρ[256U[25&2]#l]+ρ[2456U[25&2]#l]P[26],

•ρ[2568U[25&3]#l]=c[#l]ρ[256U[25&3]#l]+ρ[2456U[25&3]#l]P[26],

•r[8#l]=c[#l]r[2#l]+r[24#l],ρ[2568T[2335:l]#l|2|]=c[#l]ρ[256T[2335:l]#l|2|]+ρ[2456T[2335:l]#l|2|],

•s[258U#l]=c[#l]s[245U#l]+s[245U#l],

τ[8T[2335:l]#l]=c[#l]τ[T[2335:l]#l]+τ[4T[2335:l]#l].

Then, the following expressions are established.

PROOF1l=(C[2x[2]#l],C[2S[23]#l],(C[2S[23]#l|1|],C[2S[23]#l|2|]),C[y,C[2S[23]]#l|1|],C[2T[2&1]#l],C[2T[233]#l],(C[2T[23:l]#l|1|],C[2T[23:l]#l|2|]),C[2T[233]#l],c[25U[25&1]#l],C[2λ#l],C[2NOT(λ)#l],C[2U[25&2]#l],C[2U[25&3]#l],C[2U[235:l]#l],(C[2U[235:l]#l|1|],C[2U[235:l]#l|2|]),C[y,C[2U[235:l]]#l|1|],C[2T[2335:l]#l|2|],K[2&1|αwWh|],K[2&η|αwWh|],K[2x[2]#l],K[2J[2](x),Q[2]#l],K[2y#l],K[2r[2],s[2]#l],K[2y,C[2S[233]]#l|1|],K[2s[24S]#l],K[2r[2],T[2]#l],K[2J[2](x[2]),T[2&1]#l],K[2s[24T]#l],K[25U[25&1]#l],K[2λ,P[2]#l],K[2λ,C[2λ]#l],K[2U[25&2]#l],K[2U[25&3]#l],K[2U[235:l]#l],K[2y,C[2T[2335:l]]#l|1|],K[2s[245U]#l],c[#l],x[29&1|αwWh|]),x[29&η|αwWh|])),x[28#l],x[268#l],y[8#l],r[28#l],ρ[268S[23]#l]r[8#l]ρ[268S[233]#l|2|]s[28S#l]r[28#l]ρ[268T[2&1]#l]ρ[268T[233]#l]r[8#l]ρ[268T[233]#l|2|]ρ[2568U[25&1]#l],λ[8#l],ρ[2568λ#l],NOT(λ)[8#l],ρ[2568U[25&2]#l],ρ[2568U[25&3]#l],r[8#l]ρ[2568T[2335:l]#l|2|]s[258U#l]).

[Detail of the validity proof verification 1502 for the calculation of the first cycle]

After receiving PROOF1l, Ul+1 checks the following data.

•c[#l]=Hash(DATA1l-1,BODY1l,C[2x[2]#l],C[2S[23]#l],(C[2S[23]#l|1|],C[2S[23]#l|2|]),C[y,C[2S[23]]#l|1|],C[2T[2&1]#l],C[2T[233]#l],(C[2T[23:l]#l|1|],C[2T[23:l]#l|2|]),C[2T[233]#l],K[2s[24T]#l]c[25U[25&1]#l],C[2λ#l],C[2NOT(λ)#l],C[2U[25&2]#l],C[2U[25&3]#l],C[2U[235:l]#l],(C[2U[235:l]#l|1|],C[2U[235:l]#l|2|]),C[y,C[2U[235:l]]#l|1|],C[2T[2335:l]#l|2|],K[2&1|αwWh|],K[2&η|αwWh|],K[2x[2]#l],K[2J[2](x),Q[2]#l],K[2y#l],K[2r[2],s[2]#l],K[2y,C[2S[233]]#l|1|],K[2s[24S]#l],K[2r[2],T[2]#l],K[2J[2](x[2]),T[2&1]#l],K[25U[25&1]#l],K[2λ,P[2]#l],K[2λ,C[2λ]#l],K[2U[25&2]#l],K[2U[25&3]#l],K[2U[235:l]#l],K[2y,C[2T[2335:l]]#l|1|],K[2s[245U]#l],),

•With respect to each αwWh, the following expressions are established.

•c[#l]=c[2&1|αwWh|]+c[2&η|αwWh|],x[29&1|αwWh|] P[6B]

=c[2&1|αwWh|](C[2x[2]|αwWh|]-P[2B])+K[2&1|αwWh|],x[29&1|αwWh|] P[6B]

=c[2&η|αwWh|](C[2x[2]|αwWh|]-ηP[2B])+K[2&η|αwWh|]

●It is checked whether or not the following expressions are established.

- $x[28\#1]P[2]+x[28\#1]P[26] = c[\#1]C[2x[2]\#1]+K[2x[2]\#1]$,
- $J[2](x[28\#1])Q[2\#1] = J[2](c[\#1])Q[2\#1]+K[2J[2](x),Q[2]\#1]$,
- $y[8\#1] \in F_p$.
  $y[8\#1]P[2] = c[\#1](r[2:1]-r[2]) +K[2y\#1]$,
- $y[8\#1] \in F_p$.
- $r[28\#1]s[2:1-1] + \rho[268S[23]\#1]P[26] = c[\#1]C[2S[23]\#1] + K[2r[2],s[2]\#1]$,
- $c[\#1]C[2y,C[2S[23]]\#1|1|] + K[2y,C[2S[23]]\#1|1|] = y[8\#1]C[2S[23]\#1|2|] + \rho[268S[233]\#1|2|]P[26A[12]]$,
- $c[\#1](s[2\#1]-C[2S[233]\#1]) + K[2s[24S]\#1] = s[28S\#1]Y[:1] + \tau[8S[233]\#1]P[2]$,
- $r[28\#1]T[2:1-1] + \rho[268T[2\&1]\#1]P[26] = c[\#1]C[2T[2\&1]\#1] + K[2r[2],T[2]\#1]$,
- $J[2](x[28\#1])T[2\&1:1-1] + \rho[268T[233]\#1]P[26] = c[\#1]C[2T[233]\#1] + K[2r[2],T[2\&1]\#1]$,
- $c[\#1]C[2y,C[2T[23:1]]\#1|1|] + K[2y,C[2T[23:1]]\#1|1|] = y[8\#1]C[2T[23:1]\#1|2|] + \rho[268T[233]\#1|2|]P[26A[12]]$,

- c[#I](T[2#I]-C[2T[233]#I])
  +
  K[2s[24T]#I]
  =
  s[28T#I]Y[:I]
  -
  τ[8T[233]#I]P[2],
- r[28#I]∈A[12F_p],
- U[25:I-1]*F[25:I|0,0,0|](x[28#I])
  +
  ρ[2568U[25&1]#I]P[26]
  =
  c[#I]c[25U[25&1]#I]
  +
  K[25U[25&1]#I],
- With respect to each wWijk, i[6], j[6], k[6]
  λ[8#I|wWijk|i[6], j[6], k[6]]∈ F_p is established.

λ[8#I]P[2]+ρ[2568λ#I]P[26]=c[#I]C[2λ#I]+K[2λ,P[2]#I],λ[8#I]C[2λ#I]+ρ[2568λ#I]P[26]=c[#I]C[2λ#I]+K[2λ,C[2λ]#I],

•C[2NOT(λ)#I]=P[2]-C[2λ#I],

•c[#I]C[2U[25&2]#I]+K[2U[25&2]#I]=F[25:I|λ[8L]00|NOT(λ)[8L]00](1)*U[25&1]+ρ[2568U[25&2]#I]P[26],

•c[#I]C[2U[25&3]#I]+K[2U[25&3]#I]=F[25:I|λ[8R]00|NOT(λ)[8R]00](1)*U[25&2]+ρ[2568U[25&3]#I]P[26],

•c[#I]C[2U[235:I]#I]+K[2U[235:I]#I]=F[25:I|{00λ[8]|00NOT(λ)[8]](1)*U[25&3],

•c[#I]C[2y,C[2U[235:I]]#I|1|]+K[2y,C[2U[235:I]]#I|1|]=y[8#I]C[2U[235:I]#I|2|]+ρ[2568T[2335:I]#I|2|]P[26A[12]],

•c[#I](U[2#I]-C[2T[2335:I]#I])+K[2s[245U]#I]=s[258U#I]Y[:I]-τ[8T[2335:I]#I]P[2].

[Validity proof for the calculation of the second cycle]

[Detail of the validity proof verification 1507 for the calculation of the second cycle]

UI randomly selects the following.

$$y[4\&I] \in F_p$$

Then, PROOF2I is calculated as follows.

- P[4&I]=y[4&I]P,Θ[4&I|1|]=y[4&I]Θ[&I|1|],
- c[&I]=Hash(DATA2I-1 || BODY2I-1|| P[4&I] || Θ[4&I]),
- y[8&I]=c[&I]y[&I]+y[4&I],
- PROOF2I=P[4&I] || Θ[4&I]|| c[&I] || y[8&I].

[Detail of the validity proof verification 1502 for the calculation of the second cycle]

When receiving PROOF1I, UI+1 checks the following.

- y[8&I]P=c(R[:I]-R[:I-1])+P[4]
- y[8&I]Θ[8&I-1|0|]=c(Θ[&I-1|0|]-Θ[&I|0|])+Θ[4&I-1|0|]

Industrial Applicability

The present invention is effective at an electronic bidding, an electronic auction, or the like in the case in which the bidder and the bidding price are desired to be determined while bidding prices of bidders other than a successful bidder are kept secret and, and at the same time there is a necessity in that a third party can verify that the determination is appropriately performed, or in the case in which a third party can verify that a voting count is appropriately performed in an electronic election while anonymity is kept.

This is because with use of the present invention, if results of the above-mentioned bidding, auction, and election are made by a plurality of calculation devices, no one can newly obtain information other than the calculation result at

the same time the calculation validity can be verified by every one. Then, this calculation is more effectively performed as compared with the prior art. In addition, the number of communications performed among the calculation devices is low, so the time spent by the calculation devices to secure the communication lines is also low, which leads to the high efficiency.

CLAIMS

1. A method of calculating a value of a given function by using an apparatus including a plurality of computers, comprising:

an input process; and

an output process,

characterized in that the input process inputs a circuit and an input bit to the circuit to the plurality of computers, and

one of the computers firstly performs calculation and transmits the calculation result to another computer and the another computer which has received the calculation result performs the next calculation such that calculation is performed by one computer after another, and when all the computers have performed calculation once, the last computer which has performed calculation transmits the calculation result to the first computer which has performed calculation, and after this, calculation is performed by one computer after another and the calculation result is transmitted to the next computer such that the calculation of each cycle is repeated.

2. A method of calculating a value of a given function by using an apparatus including a plurality of computers, comprising:

an input process;

an ElGamal cipher text preparation process;

a sequential substitution reencryption process; and

a result output process,

characterized in that the input process comprises an information input step of inputting to the plurality of computers information on a circuit including a plurality of gates and information on the plurality of computers, and a dispersion input step of inputting to each of the computers each one of plural pieces of

partial data which are obtained by dispersing input data of the function into plural pieces by the number of the computers,

the ElGamal cipher text preparation process comprises an ElGamal cipher text preparation step of generating a set of ElGamal cipher texts in which at least one of the computers corresponds to the gate of the circuit that realizes the given function,

the sequential substitution reencryption process comprises a step of allowing each of the computers to perform a substitution reencryption process one after another, and the substitution reencryption process comprises a cipher text obtaining step of allowing the computer in this turn to receive the set of ElGamal cipher texts from the computer in the previous turn, a cipher text substitution and reencryption step of changing an order of the set of cipher texts received in the cipher text obtaining step for substitution and subjecting those cipher texts to reencryption, and a step of disclosing the data generated in the cipher text substitution and reencryption step to at least the computer in the next order, and

the result output process comprises a partial decryption step of deciphering or partially deciphering a part of the cipher texts generated in the cipher text substitution and reencryption step, a decryption step of deciphering a cipher text that enciphers data corresponding to the input to the circuit in the cipher texts generated in the cipher text substitution and reencryption step, and an evaluation step of evaluating an output of the circuit by using the data deciphered in the decryption step and the data partially deciphered in the partial decryption step.

3. A calculation system for evaluating a function, comprising:

a plurality of computers;

communication means for performing communication with the plurality of

computers;

input process means;

ElGamal cipher text preparation means;

sequential substitution reencryption means; and

result output means,

characterized in that the input means inputs information on a circuit whose output is desired to be obtained, information on the plurality of computers, and information on which part of an input to the circuit each of the computers has,

the ElGamal cipher text preparation means prepares ElGamal cipher texts for generating a set of ElGamal cipher texts corresponding to gates of the circuit that realizes the given function,

the sequential substitution reencryption means comprises cipher text obtaining means for allowing the computer in this turn to receive the set of ElGamal cipher texts from the computer in the previous turn, cipher text substitution and reencryption means for changing an order of the set of cipher texts received by the cipher text obtaining means for substitution and subjecting those cipher texts to reencryption, and means for disclosing the data generated by the cipher text substitution and reencryption means to at least the computer in the next order, and

the result output means comprises partial decryption means for deciphering or partially deciphering a part of the cipher texts generated by the cipher text substitution and reencryption means, decryption means for deciphering encryption related to itself of a cipher text that enciphers data corresponding to the input to the circuit in the cipher texts generated by the cipher text substitution and reencryption means, and evaluation means for evaluating an output of the circuit while using the data deciphered by the decryption means by the plurality of computers and the data partially deciphered by the partial decryption means by the plurality of computers.

4. The calculation method according to Claim 2,

characterized in that the set of ElGamal cipher texts corresponding to each of the gates is a set of ElGamal cipher texts of a secret key generated corresponding to each of the gate by each of the computers, and

a public key used for generating the ElGamal cipher texts is a sum of public keys corresponding to gates for generating two signals input to this gate.

5. The calculation method according to Claim 2,

characterized in that the input process further comprises a step of inputting an area variable of an ElGamal encryption method to each of the computers,

the ElGamal cipher text preparation process further comprises a gate secret key generating step of generating a secret key of the ElGamal cipher texts corresponding to each of the gates of the circuit by each of the computers,

each of the computers performs:

a gate public key generating step of generating a gate public key corresponding to the secret key generated in the gate secret key generating step;

a gate public key validity proof generating step of generating a gate public key validity proof for the public key generated in the gate public key generating step;

a gate public key validity proof disclosing step of disclosing the gate public key validity proof generated in the gate public key validity proof generating step;

an input gate secret key generating step of generating a secret key of the ElGamal cipher texts corresponding to a gate where an input is directly made to the circuit of the gates of the circuit;

an input gate public key generating step of generating an input gate public key corresponding to the secret key generated in the input gate secret key generating step;

an input gate public key validity proof generating step of generating a validity proof for the public key generated in the input gate public key generating step;

an input gate public key validity proof disclosing step of disclosing the input public key validity proof generated in the input gate public key validity proof generating step;

a gate public key obtaining step of obtaining gate public keys generated by other respective computers;

a gate public key integration step of integrating the gate public keys obtained in the gate public key obtaining step;

a gate public key encryption step of enciphering the gate secret key generated by this computer with the gate public key integrated in the gate public key integration step;

a gate secret key cipher text disclosing step of disclosing a gate secret key cipher text generated in the gate public key encryption step;

a gate secret key cipher text validity proof generating step of generating a validity proof for the gate secret key cipher text;

a gate secret key cipher text validity proof disclosing step of disclosing the gate secret key cipher text validity proof generated in the gate secret key cipher text validity proof generating step;

an input cipher text generating step of generating a cipher text corresponding to a part of the input of the circuit input to each of the computers.

an input cipher text validity proof generating step of generating a validity proof for the cipher text corresponding to the part of the input of the circuit generated in the input cipher text generating step;

an input cipher text validity proof disclosing step of disclosing the proof generated in the input cipher text validity proof generating step; and

an output cipher text generating step of generating and disclosing a cipher

text corresponding to an output of the gate,

the sequential substitution reencryption process comprises:

a gate secret key cipher text substitution and reencryption step of changing an order of a set of the gate secret key cipher texts with one substitution randomly selected on the basis of a predetermined permitted substitution method for reencryption;

an input cipher text substitution and reencryption step of changing an order of a set of the input cipher texts with one substitution randomly selected on the basis of a predetermined permitted substitution method for reencryption;

an output cipher text substitution and reencryption step of changing an order of a set of the output cipher texts with one substitution randomly selected on the basis of a predetermined permitted substitution method for reencryption; and

a gate secret key cipher text, input cipher text, and output cipher text substitution and reencryption validity proof generating and disclosing step of generating and disclosing validity proofs for the substitution and reencryption performed in the gate secret key cipher text substitution and reencryption step, the input cipher text substitution and reencryption step, and the output cipher text substitution and reencryption step,

the partial decryption step of the result output process comprises:

a gate secret key partial decryption step of partially deciphering the gate secret key cipher texts by mutually performing communication and calculation by the computers;

an input cipher text partial decryption step of partially deciphering the input cipher texts by mutually performing communication and calculation by the computers;

an output cipher text partial decryption step of partially deciphering the output cipher texts by mutually performing communication and calculation by the

computers; and

a gate secret key, input cipher text, and output cipher text partial decryption step validity proof generating and disclosing step of generating and disclosing the validity proofs for the partial decryption performed in the gate secret key partial decryption step, the input cipher text partial decryption step, and the output cipher text partial decryption step, and

the calculation method further comprises a step of verifying various validity proofs disclosed by other computers.

6. A calculation system, comprising a plurality of computers, input means, and output means, in which one of the computers firstly performs calculation and transmits the calculation result to another computer and the another computer which has received the calculation result performs the next calculation such that calculation is performed by one computer after another, and when all the computers have performed calculation once, the last computer which has performed calculation transmits the calculation result to the first computer which has performed calculation, and after this, calculation is performed by one computer after another and the calculation result is transmitted to the next computer such that the calculation of each cycle is repeated,

characterized in that the input means inputs information on a circuit and a part of input bits to the circuit to the computer,

the calculation of the zero-th cycle is performed before the first computer performs the calculation of the first cycle,

the plurality of computers comprise data obtaining means for obtaining transmitted data used in the calculation of each cycle, validity proof verifying means, signature text verifying means, first computer special calculating means performed by the first computer, random number generating means for performing random number generation, a main calculation calculating means for

performing a main calculation, validity proof generating means for proving a validity for a calculation performed in the main calculation, signature means, and data transmission means,

the transmitted data comprises data transmitted from other computer, data main body, a validity proof for the data main body, and a signature text,

the signature text comprises data including a signature text corresponding to a combination of the data transmitted from the other computer, the data main body, and the validity proof for the data main body,

the validity proof verifying means verifies a validity proof in the transmitted data

the signature text verifying means verifies the signature text in the transmitted data,

the main calculation calculates the random number generated by the random number generating means,

the signature means generates a signature text for a combination of the transmitted data, the data main body that is the calculation result calculated in the main calculation, and the validity proof generated by the validity proof generating means, and

the data transmission means transmits a combination of the transmitted data, the data main body that is the calculation result calculated in the main calculation, the validity proof generated by the validity proof generating means, and the signature text generated by the signature means.

7. The calculation system according to Claim 6, wherein a data main body of the transmitted data and the data main body that is the calculation result calculated in the main calculation comprise a combination of multiple sequence alignment ElGamal cipher texts on a true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring in the

calculation of the first cycle.

8. The calculation system according to Claim 7, characterized in that the calculation of each cycle comprises calculation means for the first cycle and calculation means of cycles subsequent to the first cycle,

the calculation means generates the combination of the multiple sequence alignment ElGamal cipher texts on the true value group ring and the extended multiple sequence alignment ElGamal cipher texts on the true value group ring with the calculation means of the zero-th cycle and comprises reencryption public key generating means for generating a public key used for reencryption by the calculation means of the first cycle, data conversion means for converting the transmitted data, secret key conversion means, and random number conversion means,

the data conversion means is adapted to convert the combination of the cipher texts that are the data main body with another combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring,

the secret key conversion means converts the secret key used for the combination of the cipher texts that are the calculation result of the data conversion means with a secret key corresponding to the public key generated by the reencryption public key generating means,

the calculation result of the secret key conversion means comprises a combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring,

the random number conversion means is adapted to convert a random number used for the combination of the cipher texts that are the calculation

results of the data conversion means, and

the calculation result of the random number conversion means comprises a combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring.

9. The calculation system according to Claim 8, characterized in that the calculation means of the cycles subsequent to the first cycle comprises of the calculation means of the second cycles and the calculation means of cycles subsequent to the second cycle,

the data main body of the transmitted data and the data main body calculated in the main calculation comprise a combination of multiple sequence alignment ElGamal cipher texts on the true value group ring and extended multiple sequence alignment ElGamal cipher texts on the true value group ring in the second calculation, and

the calculation means of the second cycles cipher text conversion means for converting the data main body of the transmitted data to generate an ElGamal cipher text or an ellipse curve ElGamal cipher text and partial decryption means for partially deciphering the cipher texts of the data main body of the transmitted data.

10. The calculation system according to Claim 9, characterized in that the calculation means of the cycles subsequent to the second cycle only comprises the calculation means of the third cycle,

the calculation means of the third cycle of the main calculation means outputs the transmitted data as it is, and

the validity proof generating means outputs a null string.