

【DESCRIPTION】

【Invention Title】

DEVICE PERFORMING STATISTICAL OPERATION ON HOMOMORPHIC CIPHERTEXT AND METHOD THEREOF

【Technical Field】

[0001] The disclosure relates to a device performing statistical operation on a homomorphic ciphertext and a method thereof. More particularly, the disclosure relates to an electronic device capable of effectively performing statistical operation on a homomorphic ciphertext and a method thereof.

【Background Art】

[0002] With the development of electronic and communication technology, a variety of services are being supported by utilizing data which is transmitted and received between various services. Among the examples therefrom, a user may keep one's private information or the like stored in a server, and actively use cloud computing services which uses the information in the server.

[0003] In this environment, use of security technology is essential for preventing data leakage. Accordingly, the server is configured to store encrypted data. In this case, because the server is configured to decrypt encrypted data each time when searching stored data or performing a series work based on the data, waste in resource and time may occur.

[0004] In addition, when a third-party hacking occurs while in a temporarily decrypted state for a operation in the server, there is the problem of private information being easily leaked to the third-party.

[0005] In order to solve the above-described problems and disadvantages, a

homomorphic encryption method is being researched. By using a homomorphic encryption scheme, even if an operation is performed in a ciphertext itself without decrypting encrypted information, a same result as with the encrypted value after operation on a plaintext is performed may be obtained. Accordingly, various operations may be performed on the ciphertext without performing any decryption.

[0006] However, processing time is slower than a plaintext operation scheme of the related art in that operation in a homomorphic ciphertext state requires more operational volume than the operation in the plaintext state. Specifically, the statistical operation on data requires a method which may more effectively perform a statistical operation on a homomorphic ciphertext in that much operational volume is required even when in the plaintext state.

【Disclosure】

【Technical Problem】

[0007] Aspects of the disclosure are to address at least the above-mentioned problems and/or disadvantages and to provide at least the advantages described below. Accordingly, an aspect of the disclosure is to provide an electronic device capable of effectively performing a statistical operation on a homomorphic ciphertext and a method thereof.

【Technical Solution】

[0008] According to an example embodiment, an electronic device includes a memory configured to store at least one instruction, and store homomorphic ciphertexts storing a plurality of variable data in an encrypted state in plurality, and a processor configured to execute at least one instruction, and the processor is configured to generate, by executing the at least one instruction, number data corresponding to a variable combination by using a bin mask having different variable data classified for each of the

homomorphic ciphertexts based on an operation instruction on the plurality of homomorphic ciphertexts being received.

[0009] The homomorphic ciphertext may include a plurality of slots, and each of the plurality of slots may include one variable data.

[0010] The bin mask may include a plurality of slots, and each of the plurality of slots may include data on whether one variable value is present, and the processor may be configured to generate a plurality of bin masks for each variable data included in the homomorphic ciphertext with respect to each of the homomorphic ciphertexts, select a bin mask corresponding to the variable combination from among the plurality of generated bin masks, and generate number data with the variable combination by using multiplication between the selected bin masks.

[0011] The bin mask may include a plurality of slots, and each of the plurality of slots may include a plurality of sub slots including data on whether one variable value is present, and the processor may be configured to generate one bin mask on each of the homomorphic ciphertexts, and generate number data with the variable combination by using sub slots in the bin mask which correspond to the variable combination from among the plurality of bin masks.

[0012] The plurality of sub slots may be configured to be disposed in one slot with a preset bit distance.

[0013] The processor may be configured to join a first homomorphic ciphertext and a second homomorphic ciphertext including a plurality of data on a same feature to one homomorphic ciphertext.

[0014] The processor may be configured to use a first position data in the first homomorphic ciphertext and a second position data in the second homomorphic ciphertext on common data in the first homomorphic ciphertext and the second

homomorphic ciphertext to join the first homomorphic ciphertext and the second homomorphic ciphertext as one.

[0015] The processor may be configured to compare, based on data encrypted with a one direction encryption scheme using a preset common key with respect to each of the plurality of data comprised in the first and second homomorphic ciphertexts and position data in a homomorphic ciphertext on the encrypted data being input, encrypted data on the first homomorphic ciphertext with encrypted data on the second homomorphic ciphertext, and check the first position data and the second position data which include common data between the two homomorphic ciphertexts.

[0016] According to an example embodiment, a method of processing ciphertext on a homomorphic ciphertext includes storing homomorphic ciphertexts, which stores a plurality of variable data in an encrypted state, in plurality, and receiving an operation instruction on the plurality of homomorphic ciphertexts, generating a bin mask having different variable data classified for each of the plurality of homomorphic ciphertexts, generating number data corresponding to a variable combination by using the bin mask, and outputting the generated number data.

[0017] The homomorphic ciphertext may include a plurality of slots, and each of the plurality of slots may include one variable data.

[0018] The bin mask may include a plurality of slots, and each of the plurality of slots may include data on whether one variable value is present, and the generating the bin mask may include generating a plurality of bin masks for each variable data included in the homomorphic ciphertext with respect to each homomorphic ciphertext, and the the generating number data may include selecting a bin mask corresponding to the variable combination from among the plurality of generated bin masks, and using multiplication between the selected bin masks to generate number data with the variable combination.

[0019] The bin mask may include a plurality of slots, and each of the plurality of slots may include a plurality of sub slots including data on whether one variable value is present, the generating the bin mask may include generating one bin mask with respect to each of the homomorphic ciphertexts, and the generating number data may include using sub slots in the bin mask corresponding to the variable combination from among the plurality of bin masks to generate number data with the variable combination.

[0020] The plurality of sub slots may be configured to be disposed in one slot with a preset bit distance.

[0021] The encryption processing method may further include joining a first homomorphic ciphertext and a second homomorphic ciphertext including a plurality of data on a same feature to one homomorphic ciphertext.

[0022] The joining may include using a first position data in the first homomorphic ciphertext and a second position data in the second homomorphic ciphertext on common data in the first homomorphic ciphertext and the second homomorphic ciphertext, and joining the first homomorphic ciphertext and the second homomorphic ciphertext as one.

[0023] The joining may include comparing, based on data encrypted with a one direction encryption scheme using a preset common key with respect to each of the plurality of data included in the first and second homomorphic ciphertexts and position data in a homomorphic ciphertext on the encrypted data being input, encrypted data on the first homomorphic ciphertext with encrypted data on the second homomorphic ciphertext, and checking the first position data and the second position data which include common data between the two homomorphic ciphertexts.

[0024] According to an example embodiment, a computer readable recording medium including a program for executing a ciphertext processing method includes storing

homomorphic ciphertexts, which stores a plurality of variable data in an encrypted state, in plurality, and receiving an operation instruction on the plurality of homomorphic ciphertexts, generating a bin mask having different variable data classified for each of the plurality of homomorphic ciphertexts, generating number data corresponding to a variable combination by using the bin mask, and outputting the generated number data.

【Effect of Invention】

[0025] According to various example embodiments of the disclosure as described above, various statistical processing is possible by using a homomorphic ciphertext, and statistical processing is possible by merging with respect to a homomorphic ciphertext having data structures of different schemes.

【Description of Drawings】

[0026] FIG. 1 is a diagram illustrating a structure of a network system according to an embodiment of the disclosure;

[0027] FIG. 2 is a block diagram illustrating a brief configuration of an electronic device according to an embodiment of the disclosure;

[0028] FIGS. 3 and 4 are diagrams illustrating a bin count operation method according to an embodiment of the disclosure;

[0029] FIGS. 5 and 6 are diagrams illustrating an operation of an expanded bin count operation according to an embodiment of the disclosure;

[0030] FIG. 7 is a diagram illustrating a joining method on a plurality of encryption tables;

[0031] FIG. 8 is a diagram illustrating a statistic calculation method using a bin mask

according to an embodiment of the disclosure;

[0032] FIGS. 9 and 10 are diagrams illustrating a process of generating a bin mask using plaintext according to an embodiment of the disclosure;

[0033] FIG. 11 is a diagram illustrating an approximation algorithm according to an embodiment of the disclosure;

[0034] FIG. 12 is a diagram illustrating a bin mask generating operation using a homomorphic ciphertext according to an embodiment of the disclosure;

[0035] FIG. 13 is a diagram illustrating an operation of a bin count operation according to an embodiment of the disclosure;

[0036] FIG. 14 is a diagram illustrating original data and a target of a large bin count operation according to an embodiment of the disclosure;

[0037] FIG. 15 is a diagram illustrating a method of calculating a number of a specific number of cases by using a bin mask;

[0038] FIG. 16 is a diagram illustrating an operation of a bin count operation using a power bin mask according to an embodiment of the disclosure;

[0039] FIG. 17 is a diagram illustrating a bin count operation considering an error term according to an embodiment of the disclosure;

[0040] FIG. 18 is a diagram illustrating a generation operation of a power bin mask according to an embodiment of the disclosure;

[0041] FIG. 19 is a diagram illustrating a generation operation of a power bin mask according to another embodiment of the disclosure;

[0042] FIG. 20 is a diagram illustrating an operation of multiplication operation between a plurality of bin masks according to an embodiment of the disclosure;

[0043] FIG. 21 is a diagram illustrating an operation of multiplication operation using a plurality of GPUs;

[0044] FIG. 22 is a diagram illustrating a decryption operation after multiplication operation according to an embodiment of the disclosure;

[0045] FIG. 23 is a diagram illustrating a data structure of a bin mask according to an embodiment of the disclosure;

[0046] FIG. 24 is a diagram illustrating a data structure of a multiplication operation result according to an embodiment of the disclosure;

[0047] FIG. 25 is a diagram illustrating a comparison operation according to the disclosure;

[0048] FIGS. 26 to 28 are diagrams illustrating various statistic calculation methods according to an embodiment of the disclosure;

[0049] FIG. 29 is a diagram illustrating an operation of calculating a maximum value in a slot according to the disclosure;

[0050] FIG. 30 is a diagram illustrating an operation of calculating a maximum value in several columns in a plurality of blocks;

[0051] FIG. 31 is a diagram illustrating a method of calculating a value of a specific order according to an embodiment of the disclosure; and

[0052] FIG. 32 is a flowchart illustrating a ciphertext processing method according to an embodiment of the disclosure.

【Best Mode】

[0053] -

【Detailed Description of Exemplary Embodiments】

[0054] The disclosure will be described in detail below with reference to accompanying drawings. A data transmitting process performed in the disclosure may be applied with encryption/decryption if necessary, and all expressions describing the data transmitting process in the disclosure and in the claims should be interpreted to

include encryption/decryption even if it is not specific mentioned. Expressions in forms such as “transmit (transfer) from A to B” or “receive A from B” in the disclosure may include transmitting (transferring) or receiving with another medium included therebetween, and not necessarily describe transmitting (transferring) or receiving directly from A to B only.

[0055] In describing the disclosure, an order of each step is to be understood as non-limiting unless the order of each step needs to be performed such that a preceding step must be performed logically and temporally prior to a following step. That is, except for exceptional cases as described above, even if a process described as the following step is performed preceding a process described as the preceding step, it does not influence the nature of the disclosure and the scope of protection should also be defined regardless of the order of the step. Further, in the disclosure, expressions such as “A or B” not only refers to any one of A and B selectively, but also may be defined as including both A and B. In addition, the term “include” may have a comprehensive meaning as further including another element in addition to the elements listed as included.

[0056] In the disclosure, only the essential elements necessary in describing the disclosure have been described, and elements not related to the nature of the disclosure have been omitted. Further, the disclosure is not to be construed in an exclusive sense including only the recited elements, but to be interpreted in a non-exclusive sense where other elements may be included.

[0057] Further, in the disclosure, the term “value” may be defined as not only including a scalar value, but also a vector and a polynomial form.

[0058] Mathematical operations and calculations of each step in the disclosure described below may be realized with computer operations by a coding method known for performing a relevant operation or calculation and/or coding appropriately designed

in the disclosure.

[0059] Specific equations described below are described as an example from among several possible alternatives, and the scope of protection of the disclosure should not be interpreted as being limited by the recited equations.

[0060] For convenience of description, notations such as the following will be used disclosure.

[0061] $a \leftarrow D$: select element a according to distribution D

[0062] $s_1, s_2 \in R$: each of S_1, S_2 is an element belonging to a set R

[0063] $\text{mod}(q)$: compute modular with element q

[0064] $\lfloor \cdot \rfloor$: round-off an internal value

[0065] The various example embodiments of the disclosure will be described in detail below using the accompanying drawings.

[0066] FIG. 1 is a diagram illustrating a structure of a network system according to an embodiment of the disclosure.

[0067] Referring to FIG. 1, the network system may include a plurality of electronic devices 100-1 to 100-n, a first server device 200, and a second sever device 300, and each configuration may be interconnected through a network 10.

[0068] The network 10 may be realized through a wired/wireless communication network, a broadcast communication network, an optical communication network, a cloud network, or the like of various forms, and each device may be connected in methods such as a Wi-Fi, a Bluetooth, a near field communication (NFC), or the like without a separate medium.

[0069] In FIG. 1, the electronic device has been illustrated as being in plurality 100-1 to 100-n, but the electronic devices may not necessarily be used in plurality, and one device may be used. In an example, the electronic devices 100-1 to 100-n may be

realized to devices of various forms such as a smartphone, a tablet, a game player, a personal computer (PC), a laptop PC, a home server, a kiosk, and the like, and in addition thereto, may be realized in the form of a home appliance applied with an Internet of Things (IoT) function.

[0070] The user may input various data through the electronic devices 100-1 to 100-n used by oneself. The input data may be stored in the electronic devices 100-1 to 100-n itself, but may be transmitted to an external device for reasons such as storage capacity and security and stored. In FIG. 1, the first server device 200 may perform the role of storing such data, and the second server device 300 may perform the role of using a portion or all of the data stored in the first server device 200.

[0071] Each of the electronic devices 100-1 to 100-n may homomorphically encrypt the input data, and transmit a homomorphic ciphertext to the first server device 200.

[0072] Each electronic device 100-1 to 100-n may include encrypted noise, which is calculated in the process of performing homomorphic encryption, that is, an error in the ciphertext. For example, the homomorphic ciphertext generated in each of the electronic devices 100-1 to 100-n may be generated in a form in which a result value, which includes a message and an error value when decrypting using a secret key thereafter, is stored.

[0073] In an example, the homomorphic ciphertext generated in the electronic devices 100-1 to 100-n may be generated in a form satisfying the following property when decrypting using a secret key.

[0074] [Equation 1]

[0075] $\text{Dec}(\text{ct}, \text{sk}) = \langle \text{ct}, \text{sk} \rangle = M + e \pmod{q}$

[0076] Here, $\langle$ and $\rangle$ represent a usual inner product, ct represents a ciphertext, sk represents a secret key, M represents a plaintext message, e represents an encryption

error value, and $\text{mod } q$ represents a modulus of a ciphertext. q may be selected greater than a result value M where a scaling factor(Δ) is multiplied to a message. If an absolute value of error value e is sufficiently small compared to M , a decryption value $M+e$ of the ciphertext may be a value which may substitute the original message from the significant figure operation to a same degree of precision. The error from among the decrypted data may be disposed at a least significant bit (LSB) side, and M may be disclosed at a second least significant bit side.

[0077] Based on the size of the message being too small or too large, the size may be adjusted by using a scaling factor. If the scaling factor is used, because not only the message in integer form but even the message in error form may be encrypted, utilization may be greatly increased. In addition, by adjusting the size of the message using the scaling factor, an area in which messages are present in the ciphertext after operation is performed, that is, a size of an effective area may be adjusted.

[0078] According to an example embodiment, a ciphertext modulus q may be set to various forms and used. In an example, the modulus of the ciphertext may be set to a form of an exponentiation $q=\Delta^L$ of the scaling factor Δ . If Δ is 2, it may be set to a value such as $q=2^{10}$.

[0079] In another example, the ciphertext modulus may be set to a value of which a plurality of different scaling factors are multiplied. Each factor may be set to a value within a similar range, that is, a value of a size similar with one another. For example, it may be set to $q=q_1 q_2 q_3 \dots q_x$, and each of $q_1, q_2, q_3, \dots, q_x$ may be a size similar with scaling factor Δ , and may be set to a value of a small relationship with one another.

[0080] If the scaling factor is set in this method, because a whole operation can be carried out by separating into a plurality of modulus operations according to a Chinese Remainder Theorem (CRT), the burden of operation may be reduced.

[0081] In addition, by using factors of similar sizes with one another, when performing rounding in a step to be described below, a value nearly similar with the result value of the previous example may be obtained.

[0082] The first server device 200 may not decrypt the received homomorphic ciphertext, and store in the ciphertext state.

[0083] The second server device 300 may be configured to request a specific processing result on the homomorphic ciphertext to the first server device 200. The first server device 200 may be configured to transmit, after performing a specific operation according to a request of the second sever device 300, the result to the second sever device 300. Here, the specific operation may not only be general operations such as performing addition on a plurality of homomorphic ciphertexts and homomorphic multiplications, but also operations such as a statistical operation, for example, an average, a frequency distribution, a linear regression, a covariance, or the like.

[0084] At this time, the second server device 300 may be configured to perform a joining operation on the plurality of homomorphic ciphertexts.

[0085] In an example, based on ciphertexts ct_1 and ct_2 transmitted by the two electronic devices 100-1 and 100-2 being stored in the first server device 200, the second server device 300 may be configured to request a value of aggregated data provided from the two electronic devices 100-1 and 100-2 to the first server device 200. The first server device 200 may be configured to transmit, after performing operation of aggregating the two ciphertexts according to the request, the result value ($ct_1 + ct_2$) to the second sever device 300.

[0086] Based on the properties of the homomorphic ciphertext, the first server device 200 may be configured to perform operation in a state not having performed decryption, and the result value thereof may be in ciphertext form. At this time, the first server

device 200 may be configured to perform bootstrapping on the operation result.

[0087] The first server device 200 may be configured to transmit an operation result ciphertext to the second sever device 300. The second sever device 300 may be configured to decrypt the received operation result ciphertext and obtain the operation result value of data included in each homomorphic ciphertext. Further, the first server device 200 may be configured to perform operation according to a user request numerous times.

[0088] In FIG. 1, encryption being performed in the first electronic device and the second electronic device, and the second server device performing decryption have been illustrated, but the embodiment is not limited thereto.

[0089]

[0090] FIG. 2 is a block diagram illustrating a configuration of an electronic device according to an embodiment of the disclosure.

[0091] Referring to FIG. 2, the electronic device 100 may include a memory 110, a processor 120, a communication device 130, a display 140, and an operation input device 150. The electronic device described above may be various devices such as a personal computer (PC), a notebook, a smartphone, a tablet, a server, or the like.

[0092] The memory 110 may be configured to store at least one instruction on the electronic device 100. For example, the memory 110 may be stored with various programs (or software) for the electronic device 100 to operate according to the various example embodiments of the disclosure.

[0093] The memory 110 as described above may be realized to various forms such as a random access memory (RAM) or a read only memory (ROM), a Buffer, a cache, a flash memory, a hard disk drive (HDD), an external memory, a memory card, or the like, and is not limited to any one.

[0094] The memory 110 may be configured to store a message to be encrypted. Here, the message may be various credit data, private data, or the like variously cited by the user, and may be data associated with use history, or the like such as position data and internet use time data used in the electronic device 100.

[0095] Further, the memory 110 may be configured to store a public key, and store, based on the electronic device 100 generating the public key directly, not only a secret key, but also various parameters required in generating the public key and the secret key.

[0096] Further, the memory 110 may be configured to store the homomorphic ciphertext generated in a process described below. Further, the memory 110 may be configured to store the homomorphic ciphertext transmitted from the external device. In addition, the memory 110 may be configured to store the operation result ciphertext which is a result product of an operation process described below.

[0097] The communication device 130 may be formed to connect the electronic device 100 with the external device (not shown), and may be formed not only in a form connecting to the external device through a local area network (LAN) and an internet network, but also in a form connecting through a universal serial bus (USB) port or a wireless communication (e.g., WiFi 802.11a/b/g/n, NFC, Bluetooth) port. The communication device 130 may be referred to as a transceiver.

[0098] The communication device 130 may be configured to receive the public key from the external device, and transmit the public key generated on its own by the electronic device 100 to the external device.

[0099] Further, the communication device 130 may be configured to receive a message from the external device, and transmit the generated homomorphic ciphertext or the operation result to the external device.

[00100] In addition, the communication device 130 may be configured to

receive various parameters required in generating the ciphertext from the external device. The various parameters upon realization may be received directly from the user through the operation input device 150 which will be described below.

[00101] In addition, the communication device 130 may be configured to receive a request of operation on the homomorphic ciphertext from the external device, and transmit the calculated result according thereto to the external device. The requested operation may be operation such as addition, subtraction, and multiplication (e.g., modular multiplication operation), and may be statistical operation. Here, modular multiplication operation may refer to modular operation with a q element.

[00102] The display 140 may be configured to display a user interface window for selecting a function supported by the electronic device 100. For example, the display 140 may be configured to display the user interface window for selecting various functions provided by the electronic device 100. The display 140 may be a monitor such as a liquid crystal display (LCD), an organic light emitting diodes (OLED), or the like, and may be realized to a touch screen capable of simultaneously performing a function of the operation input device 150 which will be described below.

[00103] The display 140 may be configured to display a message requesting input of a parameter required in generating a secret key or a public key. Further, the display 140 may be configured to display a message having the subject of encryption to select the message. The subject of encryption upon implementation may be selected directly by the user, or selected automatically. That is, private data and the like required in encryption may be set automatically even if the message is not directly selected by the user.

[00104] The operation input device 150 may be configured to receive input of a function selection of the electronic device 100 and a control command on a relevant

function from the user. For example, the operation input device 150 may be configured to receive a parameter required in generating the secret key and the public key from the user. In addition, the operation input device 150 may be configured to receive, from the user, the setting of the message to be encrypted.

[00105] The processor 120 may be configured to control the overall operation of the electronic device 100. For example, processor 120 may be configured to control, by executing at least one instruction stored in the memory 110, the operation of the electronic device 100 overall. The processor 120 may be configured to a single device such as a central processing unit (CPU) and an application-specific integrated circuit (ASIC), or configured to a plurality of configurations such as the CPU and a graphics processing unit (GPU).

[00106] When the message to be transmitted is input, the processor 120 may be configured to store in the memory 110. Further, the processor 120 may be configured to use the various setting values and program stored in the memory 110 to homomorphically encrypt the message. In this case, the public key may be used.

[00107] The processor 120 may be configured to use a public key required in performing encryption by generating the public key on its own, or may and receive and use from the external device. In an example, the second sever device 300, which performs decryption, may be configured to distribute the public key to other devices.

[00108] When generating the key on its own, the processor 120 may be configured to generate the public key by using a Ring-LWE technique. For example, the processor 120 may be configured to first set various parameters and ring, and store in the memory 110. An example of the parameter may be a length of a plaintext message bit, a size of the public key and the secret key, and the like.

[00109] The ring may be represented with Equation 2 as below.

[00110] [Equation 2]

[00111]
$$R = \frac{Z_q[X]}{f(x)}$$

[00112] Here, R represents the ring, Z_q represents a coefficient, $f(x)$ represents an n-th polynomial.

[00113] The ring, as a set of polynomials having a predetermined coefficient, may refer to a set of which addition and multiplication between the elements are defined and closed with respect to the addition and multiplication. The ring may be referred to as a ring.

[00114] In an example, the ring may refer to a set of an n-th polynomial where the coefficient is Z_q . For example, if n is $\Phi(N)$, it may mean an N-th cyclotomic polynomial. ($f(x)$) may represent an ideal of $Z_q[x]$ which is generated as $f(x)$. A Euler totient function $\Phi(N)$ may refer to a number of natural numbers disjoint from N and smaller than N. When $\Phi_N(x)$ is defined as the N-th cyclotomic polynomial, the ring may be represented with Equation 3 as below. Here, 2^{17} may be used for N.

[00115] [Equation 3]

[00116]
$$R = \frac{Z_q[X]}{\Phi_N(x)}$$

[00117] The secret key (sk) may be represented as below.

[00118] The ring of Equation 3 described above may include a complex number in the plaintext space. In order to increase operational speed on the homomorphic ciphertext, only the set of which the plaintext space is a real number from among the above-described set of rings sets may be used.

[00119] When the ring as described above is set, the processor 120 may be configured to calculate the secret key (sk) from the ring.

[00120] [Equation 4]

[00121]
$$sk \leftarrow (1, s(x)), s(x) \in R$$

[00122] Here, $s(x)$ may refer to a polynomial randomly generated as a small coefficient.

[00123] Further, the processor 120 may be configured to calculate a first random polynomial($a(x)$) from the ring. The first random polynomial may be represented as below.

[00124] [Equation 5]

[00125] $a(x) \leftarrow R$

[00126] In addition, the processor 120 may be configured to calculate an error. For example, the processor 120 may be configured to calculate an error from a discrete Gaussian distribution or a distribution having a close statistical distance therefrom. The error may be represented as below.

[00127] [Equation 6]

[00128] $e(x) \leftarrow D_{\alpha q}^n$

[00129] When even the error is calculated, the processor 120 may be configured to perform a modular operation of the error to the first random polynomial and the secret key to calculate a second random polynomial. The second random polynomial may be represented as below.

[00130] [Equation 7]

[00131] $b(x) = -a(x)s(x) + e(x) \pmod{q}$

[00132] Finally, the public key (pk) may be set as below in a form which includes the first random polynomial and the second random polynomial.

[00133] [Equation 8]

[00134] $pk = (b(x), a(x))$

[00135] Because the above-described key generation method is merely one example, the embodiment is not necessarily limited thereto, and the public key and the

secret key may be generated in other methods in addition to the above.

[00136] The processor 120 may be configured to control, based on the public key being generated, the communication device 130 to transmit to the other devices.

[00137] Further, the processor 120 may be configured to generate the homomorphic ciphertext on the message. For example, the processor 120 may be configured to apply the public key generated previously on the message to generate the homomorphic ciphertext.

[00138] The message to be decrypted may be received from an external source, and may be input from an input device directly included in or connected to the electronic device 100. For example, based on the electronic device 100 including a touch screen or a key pad, the processor 120 may be configured to store data input through the touch screen or the keypad by the user in the memory 110, and then encrypt the input data. The generated homomorphic ciphertext may be in a form which is restored to a result value of adding the error to a value which reflects the scaling factor in the message when performing decryption. The scaling factor may use a value, which is previously input and set, as is.

[00139] Alternatively, the processor 120 may be configured to perform encryption by using the public key immediately while multiplying the message and the scaling factor. In this case, the error calculated in the encryption process may be added to the result value of multiplying the message and the scaling factor.

[00140] In addition, the processor 120 may be configured to generate a length of the ciphertext to correspond to a size of the scaling factor.

[00141] Further, the processor 120 may be configured to control, based on the homomorphic ciphertext being generated, the communication device 130 to store in the memory 110, or transmit the homomorphic ciphertext to another device according to a

user request or a pre-set default instruction.

[00142] According to an example embodiment of the disclosure, packing may be performed. When packing is used in a homomorphic encryption, it may be possible to encrypt multiple messages to one ciphertext. In this case, when an operation is performed between each of the ciphertexts in the electronic device 100, because consequentially operations on multiple messages are processed in parallel, the operational burden is greatly reduced.

[00143] For example, the processor 120 may be configured convert, based on the message being formed of a plurality of message vectors, the plurality of message vectors to a polynomial of a form which may be encrypted in parallel, and then perform homomorphic encryption by multiplying the scaling factor to the polynomial and using the public key. Accordingly, the processor 120 may be configured to generate the ciphertext which carried out packing of the plurality of message vectors.

[00144] Based on the data stored by the electronic device 100 being a statistical table, the processor 120 may be configured to generate, in the generating process of the homomorphic ciphertext, the homomorphic ciphertext including variable data in a plurality of slots in the ciphertext. In addition, the processor 120 may be configured to generate, in the generating process of the homomorphic ciphertext, a bin mask on the relevant homomorphic ciphertext. The specific bin mask generating operation will be described below with reference to FIG. 3.

[00145] Further, the processor 120 may be configured to apply, based on decryption being required on the homomorphic ciphertext, the secret key to the homomorphic ciphertext to generate a decryption text of a polynomial form, and generate a message by decoding the decryption text of the polynomial form. The message generated at this time may include an error as described in Equation 1

described above.

[00146] Further, the processor 120 may be configured to perform an operation on the ciphertext. For example, the processor 120 may be configured to not only perform an operation of addition, subtraction, multiplication, or the like while maintaining the encrypted state on the homomorphic ciphertext, but also perform various statistical operations such as an average and frequency distribution on a plurality of data. The specific statistical operation method will be described below with reference to FIG. 3.

[00147] The electronic device 100 may be configured to detect, based on the operation being completed, data of an effective area from the operation result data. For example, the electronic device 100 may be configured to perform a rounding process of the operation result data to detect data of the effective area.

[00148] Here, the rounding process may mean proceed with a round-off of the message in the encrypted state, and may otherwise be referred to as rescaling. For example, the electronic device 100 may be configured to remove a noise area by multiplying a reciprocal number Δ^{-1} of the scaling factor to a component of each of the ciphertexts and rounding-off. The noise area may be set to correspond to the size of the scaling factor. Consequentially, a message of an effective area with the noise area excluded may be detected. Because it is proceeded in the encrypted state, additional errors may be generated, but because the size is sufficiently small, it may be disregarded.

[00149] Further, in the above-described rounding process, the modular multiplication operation as described above may be used.

[00150] In addition, the electronic device 100 may be configured to expand, based on a weight of an approximate message in the operation result ciphertext exceeding a threshold value, the plaintext space of the operation result ciphertext. For

example, if q is smaller than M in the above-described Equation 1, because $M+e \pmod q$ is to have a different value from $M+e$, decryption may not be possible. Accordingly, the value of q is to be maintained greater than M at all times. However, the value of q is gradually decreased as the operation proceeds. The expansion of the plaintext space may refer to changing ciphertext ct to a ciphertext having a greater modulus. The operation of expanding the plaintext space may otherwise be referred to as rebooting. In performing rebooting, the ciphertext may be in a state in which operation is possible once again.

[00151] The electronic device 100 according to the disclosure as described above may not only effectively perform an operation on the homomorphic ciphertext, but also on a complex statistical operation. In addition, the electronic device 100 may be configured to manage homomorphic ciphertext provided from multiple devices in one database (DB).

[00152] The specific operation of statistical operation on the homomorphic ciphertext will be described below.

[00153] First, for an effective statistical operation on the homomorphic ciphertext, the homomorphic ciphertext may be generated to include the data structure as described below.

[00154] Described is which method to store table data of the plaintext, that is, multiple record data formed of various features in the homomorphic encryption which provides a Single Instruction Multiple Data (SIMD) function. To this end, data may be gathered for each feature and stored in the ciphertext. That is, one ciphertext may store only data belonging to one feature. Here, this means that one ciphertext may store a plurality of variable values on one feature and not one ciphertext including only one data.

[00155] The homomorphic ciphertext may include multiple slots, and each slot may store multiple data. Accordingly, using the above, values on one feature (i.e., same column data in a table) may be stored in each of the multiple slots.

[00156] Specifically, table data may be stored and managed in the form as below. For example, if the size of the plaintext table is n and m (here, n is length of data row, and m is length of data column (=number of features)), and the number of data which may include a fully homomorphic ciphertext is $M(=N/2)$, the encryption table including the encrypted data may include the description as below.

[00157] 1. ciphertext

$C_{0,0}, C_{0,1}, \dots, C_{0, \lceil n/M \rceil}, C_{1,0}, \dots, C_{m-1,0}, C_{m-1,1}, \dots, C_{m-1, \lceil n/M \rceil}$ (here, ciphertext C_{ij} is (j-1)-th ciphertext including i+1-th feature, and the number of ciphertexts is $\lceil n/M \rceil$ for each feature,

[00158] 2. the number of the total features is m , data column is n ,

[00159] 3. the number of data comprised in other additional metadata (e.g., names of each feature, one ciphertext (also referred to as 1 Block), and table name).

[00160] Based on using the method as described above, a more effective calculation may be possible when calculating the statistical value for each floating feature in the encrypted table data. In addition, table joining may be effectively performed without decryption with respect to encryption tables of different methods. The table joining operation will be described below with reference to FIG. 7.

[00161] Each homomorphic ciphertext including data on one feature has been assumed below, and an operation on statistical operation with respect to homomorphic ciphertext will be described.

[00162] First, the homomorphic ciphertext may perform an operation while in an encrypted state, but because the operation of the encryption process consumes much

time, an efficient operation method is required.

[00163] In terms of statistical processing, it is first necessary to find a variable which coincides with a specific condition, and find the variable which preferentially coincides with the specific condition in that processing such as an average and dispersion on the found variable are performed.

[00164] In this aspect, a bin count operation for finding a variable which satisfies the specific condition will be first described below with reference to FIGS. 3 and 4.

[00165]

[00166] FIGS. 3 and 4 are diagrams illustrating a bin count operation method according to an embodiment of the disclosure. Here, the bin count operation may be an operation performing a role of counting the number of possible cases on a combination of each of the variable values in the data by inputting data of two or more bin variables.

[00167] Referring to FIG. 3, bin data 310, intermediate data 320, and result data 330 are illustrated. For example, based on bin values in each of the three variables (A, B, C) being {1, 2, 3, 4}(A), {1, 2, 3, 4}(B), {1, 2}(C) respectively in FIG. 3, intermediate data according to the combination of the three variables may be generated.

[00168] After generating intermediate data as described above, the result data 330 representing possible combinations of the three variables and a number on the relevant combinations may be generated by using the intermediate data with a count value.

[00169] The detailed operation of generating the intermediate data described above will be described below with reference to FIG. 4.

[00170] Referring to FIG. 4, first is the process of calculating the bin count when the values of variable A, B, C are each 1.

[00171] Each of the variables 410, 420 and 430 may include the plurality of slots. In the illustrated example, although it has been illustrated as having eight slots, slots of more than or less than nine may be included at realization.

[00172] A bin mask on each variable value may be generated with respect to each of the variables. For example, in case of variable A 410, because four values are included, a bin mask 420 corresponding to each of the four values may be formed. Further, in case of variable B 430, because four values are included, the bin mask 420 on each of the four values may be formed, and in case of variable C 450, because two values are included, a bin mask 460 on each of the two values may be formed.

[00173] The bin mask as described above may be formed at the time of encryption or may form a bin mask by calculating after being encrypted using a table look-up table function. A more detailed operation of generating the bin mask will be described with reference to FIGS. 9 to 12.

[00174] Then, when checking a bin count relevant to a specific combination, the bin mask corresponding to the relevant combination may be selected, and a result may be checked through the multiplication of the selected bin masks. For example, when checking of 1, 1, 1 combination is required, an operation of multiplying a bin mask 471 of value 1 from among the bin masks on variable A 420, a bin mask 472 of value 1 from among the bin masks on variable B 430, and a bin mask 473 of value 1 from among the bin masks on variable C 450 with one another may be performed.

[00175] An output mask 480 generated by the operation as described above may include a value of 1 at a position relevant to the relevant combination. Through the above, the position in each variable which form the relevant combination and the number of relevant combinations may be checked.

[00176] When expanding the operation as described above, a bin average and a

bin variance may be calculated. The bin average may be an operation of obtaining an average on another variable of data having the combination of the specific bin variable. Further, the bin variance may be an operation of obtaining a variance on data. A method of performing the relevant operation may include multiplying the bin mask value on the intermediate data 320 in FIG. 3 with a value of another variable of which an average is sought and just by picking only values on rows having the desired combination of bin values, all the values may be added. Then, the bin mask value may be all added, and an average may be obtained by dividing with the latter value from the former value. By utilizing the above-described method, calculation for variance may also be performed. A more detailed operation on the statistical operation will be described with reference to FIGS. 26 to 30.

[00177] The bin count described above may be used in the classification process, and may be used in methods such as association rule mining. However, in order to raise accuracy in classification, many number of possible cases in classification is required, and a data analyzer may operate bin count by combining many variables having a greater range in bin value.

[00178] Data of various continuous values may be represented as quantile data for convenience of analysis, and as an example, there may be fifty variable values with respect to one.

[00179] If there are fifty variable values with respect to one variable, the number of possible cases is significant. When the number of possible cases is significantly increased, if bin count is performed by generating the bin mask for each variable value, the total number of operations may be $(n-1) \times w \times \lfloor u/M \rfloor$ (here, n is the number of variables, w is the number of combinations, M is the number of slots of the ciphertext, and u is the length of a row of whole data). Here, if w and u are a unit of

millions, n is less than or equal to 10, and M is a unit of tens of thousands, when the number of multiplication times required reaches billions, the necessary time required be dozens of days or more time even if the time for performing multiplications is set as several milliseconds.

[00180] Accordingly, a modified method of bin count in which types of variables are varied will be described below.

[00181] The bin mask below is represents the bin value in an encoded form. For example, when there is data having a range of $[1,10]$ bin value, the bin value may be represented as 10 bytes. If representing bin value $i(\in[1,10])$ is desired, a method of setting i -th byte as 1 from among ten bytes and the rest as 0 may be used.

[00182] The bin masks set in this method may be added between one another. The above will be described with reference to FIG. 5.

[00183]

[00184] FIG. 5 is a diagram illustrating an operation of expanded bin count operation according to an embodiment of the disclosure.

[00185] Referring to FIG. 5, one power bin mask 502 and 512 may include the plurality of slots, and each of the plurality of slots may include a plurality of sub slots. Because a first variable 501 has four different variable values, a first power bin mask 502 may include four sub slots, and each sub slot may include data on whether a specific variable value is present. The generating operation of the power bin mask will be described below.

[00186] When utilizing the power bin mask as described above, individual bin masks may have been generated for each variable type in the previous process, but the power bin mask may be satisfied with one for each variable type.

[00187] Further, the operation method may be the same as the bin mask method,

and if a specific combination is required, output data 520 may be generated by using the sub slot of the power bin mask corresponding to the relevant combination. Then, output data 520 may be decoded, and the value of each combination may be calculated 530.

[00188] If the method as shown in FIGS. 3 and 4 is on generating an individual bin mask for each variable value, the method of FIG. 5 is on utilizing the sub slot which divided the slot as in FIGS. 3 and 4, and is in a form totaling the plurality of bin masks to one.

[00189] Advantages of utilizing the power bin mask as described above will be described with reference to FIG. 6.

[00190] Referring to FIG. 6, a bin operation may be performed on two variables 601 and 604 having one to four bin values, and when an existing bin mask is used, sixteen multiplications may be required. Accordingly, a final result may be known after performing decryption on sixteen ciphertexts. On the other hand, when an expanded bin mask 602 is used, the result may be obtained with only four multiplications and four decryptions. This may be because all position data is included based on the variable in the multiplication result being between 1 and 4.

[00191] Because the multiplication between the expanded bin masks is not operated based on the feature of the proposed method, only one expanded bin mask may participate in the calculation of the bin count. In addition, because the number of bins which may be included in the one bin mask is 50 at maximum, a reduction of multiplication times which may be improved by using the proposed method may be $1/50$. In addition, because the multiplication which uses the expanded bin mask is to include multiple bin data in one slot, and this results in the increase of bit length to be managed per slot, this may also influence the multiplication time. The number of bins and number of reduction times described above are merely exemplary, and the

numerical value may be changed according to a homomorphic encryption scheme applied.

[00192] For the statistical operation, it may be necessary to use not only a table of one owner, but also tables of different owners.

[00193] Accordingly, a method of joining tables owned by different devices to one table will be described below.

[00194] Different databases (DBs) may store homomorphically encrypted data in a different method. However, when each of the different DBs is stored by dividing by feature and dividing by feature of the table, it is possible to easily join the two tables.

[00195] Specifically, it may be assumed that an encrypted table is owned by a first electronic device 100-1 and a second electronic device 100-2 which are different, and key data for joining is shared therebetween. If there is a third electronic device 100-3 honestly performing protocol, two encrypted tables may be joined with the assistance of the third electronic device 100-3. The method as described above may be effectively performed in that data joining is possible without an additional homomorphic encryption operation excluding table encryption. In addition, even from a security aspect, there is the advantage that other than the number (inner-join) of common data in the process of performing joining of the plurality of tables or the number (outer-join) of data not owned by oneself from among the data owned by a counterpart, no other data is exposed.

[00196] To this end, column values relevant to the key for joining may perform one-way encryption where decryption is impossible by using a separate hash-based message authentication code (HMAC) function rather than homomorphic encryption. At this time, because each of the electronic devices 100-1 and 100-2 share the same HMAC key, the HMAC performance result on the same key value may be the same.

However, because the HMAC result also rely on the shared key value, even if it is the same key value, when another HMAC key is used, another HMAC value may be obtained.

[00197] The HMAC value and a position value of the row in the table of the original join key used for forming the relevant HMAC value may be combined to form a pair of data, and this data set may be arranged by using the HMAC key value and then transmitted to the third electronic device 100-3. Here, the position value may refer to the join key value and a row number of the other data connected therewith.

[00198] A joining company may be configured to identify keys which match by using the HMAC value, based on each electronic device sending row value data gathered together, the relevant first electronic device and second electronic device may be configured to transmit data which encrypted the relevant rows to the third electronic device 100-3, and the third electronic device 100-3 may be configured to perform joining by combining metadata of the data sent from the two devices and forming metadata of one joined table.

[00199] A more detailed joining operation will be described below with reference to FIG. 7.

[00200]

[00201] FIG. 7 is a diagram illustrating a joining method on a plurality of encryption tables.

[00202] Referring to FIG. 7, the two electronic devices 100-1 and 100-3 may respectively own tables with values of rows and columns different from one another, and if a join key (e.g., resident identification number, or the like in case there are two tables containing data related to human life) which is close to 1:1 (one-to-one) joining that may join the two tables is present, it may mean a protocol forming an encryption

table of 1) form on a joining table which joins the two tables.

[00203] Participants of the protocol are as follows. Data owning companies D_1 and D_2 may be present, a data joiner F , and Z which will own the finally encrypted joining table may be present. Here, D_1 and D_2 may be the electronic device or server illustrated in FIG. 1. In addition, the data joiner may also be the electronic device or server illustrated in FIG. 1.

[00204] The data owning companies D_1 and D_2 may have the same homomorphically encrypted encryption key instance (pk_h), and also share a same MAC key (symmetric key sk_{MAC} : 256 bit random bit). The data analyzing company Z may have a calculation key evk_h which may perform calculation on data encrypted with pk_h .

[00205] A parameter and algorithm on homomorphic encryption may be shared by D_1 , D_2 , and Z , and the MAC algorithm ($MAC: \{0,1\}^* \times \{0,1\}^{|sk_{MAC}|} \rightarrow \{0,1\}^{MAC_{len}}$) may be shared by A and B.

[00206] Data owned by the data owning companies D_1 and D_2 may be described as below.

[00207] The data owning companies D_1 and D_2 may have data of $m_{D_1} \times n_{D_1}$ ($m_{D_2} \times n_{D_2}$) size each, and a first feature may represent IDs (used as join key) of owners of each data, and may be represented as $id^{D_{10}}, \dots, id^{D_{1m_{D_1}-1}}$ ($id^{D_{20}}, \dots, id^{D_{2m_{D_2}-1}}$).

[00208] The features in the remaining columns may be represented as $f^{A_1}, \dots, f^{A_{n_{A_1}-1}}$ ($f^{B_1}, \dots, f^{B_{n_{B_1}-1}}$).

[00209] In addition, $f^X(id^X_k)$ ($i \in [0, m_X-1], k \in [0, n_X-1], X \in \{D_1, D_2\}$) may refer to a value of f_i feature on id_k user owned by company X . (Here, $[a, b]$ is a set of all integers which is greater than a and smaller than or equal to b .)

[00210] Accordingly, a data tuple on an arbitrary user $id^{D_{1_i}}$ owned by D_1 company may be defined as $(id^{D_{1_i}}, f_1^{D_1}(id^{D_{1_i}}), \dots, f_{m_{x-1}}^{D_{1_{m_{x-1}}}}(id^{D_{1_i}}))$.

[00211]

[00212] An operation may be divided into an inner-join of performing joining with respect to the common key present in the data owning companies D_1 and D_2 , and an outer-join of performing joining with respect to all data present in the two companies. The detailed operation is described as below.

[00213]

[00214] (Initial environment) Companies D_1 , D_2 , F , and Z are present, and the data owned by the respective companies are the same as described above.

[00215] 1. Each company D_X ($\in \{1, 2\}$) may add a new column and

perform $MAC_{sk_{MAC}}(id_i^{D_X})$ to the relevant columns with respect to all i and X . The relevant value may be added to the row represented as id^{X_i} . The relevant value may be described as $mac_{id^{X_i}}$.

[00216] 2. All data may be arranged by a row unit based on $mac_{id^{X_i}}$ value. The arranged MAC value may be described as mac^{X_i} , and the ID value in the relevant position may be described as id^{X_i} . That is data after arrangement may be stored in

order in $(mac^{X_i}, id^{X_i}, f_1^X(id^{X_i}), \dots, f_{m_{x-1}}^{X_{m_{x-1}}}(id^{X_i}))$ ($i \in [0, n_X - 1]$) form. That is, the row

including mac_i may be recorded in the i -th row of an input table. The table may be referred to as T_X . T_X may be owned by D_X .

[00217] 3. Each company D_X may transfer $L_X = (i, mac_i^{X_i}) : i \in [0, m_{D_X} - 1]$ to F so that 1 matches the order.

[00218] 4. F may use the received L_1, L_2 to calculate

$S_{D_1} = \{mac_i^{D_1} | (i, mac^{D_1}) \in L_1\}$, $S_{D_2} = \{mac_i^{D_2} | (i, mac^{D_2}) \in L_2\}$, and then define as

$S_{D_1 \cap D_2} = \{mac^{D_1} | mac^{D_1} \in S_{D_1} \text{ and } \exists mac^{D_2} \in S_{D_2} \text{ such that}$

$mac^{D_1} = mac^{D_2}\}$. In addition, define as $S_{D_1 - D_2} = S_{D_1} - S_{D_1 \cap D_2}$, $S_{D_2 - D_1} = S_{D_2} - S_{D_1 \cap D_2}$.

If function $index^X: S_{D_x} \rightarrow \mathbb{Z} \cup \{\perp\}$ is defined, it may be defined as

$index^X(mac^{D_x}) = i$ if $(i, mac^{D_x}) \in L_x$ otherwise $\perp$ with respect to

$mac^{D_x} \in S_{D_x}$. If $index^X(S_{D_x})$ performs $index^X$ function respectively on all

elements of S_{D_x} , it may be defined as converting the result to a data form with order.

[00219] First, the outer-join will be first described.

[00220] a) Data with order $P_{D_1} = (S_{D_1 - D_2}, S_{D_1 \cap D_2})$, $P_B = (S_{D_1 \cap D_2}, S_{D_1 - D_2})$

are formed.

[00221] b) $R_{D_1} \leftarrow index^{D_1}(P_{D_1})$, $R_{D_2} \leftarrow index^{D_2}(P_{D_2})$ are performed.

[00222] c) A random number seed s_{D_1}, s_{D_2} is generated to use permutation

π . Using the same, the values of R_{D_1} , R_{D_2} are mixed by performing

$\pi_{s_{D_1}}(R_{D_1}) \rightarrow UID_1$, $\pi_{s_{D_2}}(R_{D_2}) \rightarrow UID_2$. Here, $\pi_s(R)$ may mean a safe

permutation which mixes the order of R value using randomness provided by s , and

if s is unknown, data on data R with the original order may not be learned. In

addition, if s is known, $\pi^{-1_s}()$ may be performed.

[00223] d) F may transfer UID_1 to D_1 , and transfer UID_2 to D_2 . In addition,

F may transfer $|S_{D_2 - D_1}|$ to D_1 , and transfer $|S_{D_2 - D_1}|$ to D_2 .

[00224] e) The D_x which received UID_x may arrange data matching the

order of values in the relevant sequence. That is if $UID_X = \{i_0, i_1, \dots, i_{n_{UID_X}}\}$, data may

be arranged in $D_X = (f_1^X(id^{i_0}), \dots, f_{n_{UID_X}}^X(id^{i_{n_{UID_X}}}))_{i=i_0, i_1, \dots, i_{n_{UID_X}}}$ order. Here, it may mean

$$n_{UID_X} = |UID_X|$$

[00225] f) After, $\perp$ sequence of length m_{D_X} formed of $(\perp, \dots, \perp)$ may be added to D_X by $|S_{D_X - (\{D_1, D_2\} - D_X)}|$ number.

[00226] g) X may use the method in 1) to encrypt D_X by column to form an encryption table object C^{tb_x} , and then transfer with the metadata to F.

[00227] h) F may use $\pi^{-1_{s_{D_1}}}, \pi^{-1_{s_{D_2}}}$ to restore the order of C^{tb_1}, C^{tb_2} received from each of D_1, D_2 . Here, the role of $\pi^{-1_{s_{D_1}}}, \pi^{-1_{s_{D_2}}}$ may be performing the role of receiving input of the order of each element of C^{tb_1}, C^{tb_2} and sending the whole encrypted element to a row position which corresponds to the result number. At this time, the ciphertexts on the data added in g) step do not undergo this process.

[00228] i) If the result of h) is C'_1, C'_2 respectively, the two tables may be combined and one encrypted table C' may be formed. At this time, metadata of the joined encryption table may be generated by joining the metadata of each of C'_1, C'_2 .

[00229] j) F may transfer the joined encryption table C_R to Z .

[00230] The inner-join may be similar to the outer-join above, but is different in a), b) and d) processes as below. In addition, f) process may not be performed.

[00231] a) Data with order $P_{D_1 \cap D_2} = (S_{D_1 \cap D_2})$ may be formed.

[00232] b) $R_{D_1} \leftarrow \text{index}^{D_1}(P_{D_1 \cap D_2}), R_{D_2} \leftarrow \text{index}^{D_2}(P_{D_1 \cap D_2})$ may be

performed.

[00233] d) F may transfer UID_1 to D_1 , and UID_2 to D_2 .

[00234] An actual statical operation using the bin mask described above will be

described below.

[00235]

[00236] FIG. 8 is a diagram illustrating a statistic calculation method using a bin mask according to an embodiment of the disclosure.

[00237] Referring to FIG. 8, raw data and numerical data on the relevant raw data may be shown. Because the operation of converting numerical data with respect to raw data is common, the detailed description will be omitted. In the illustrated example, the numerical data has been shown in a plaintext state for convenience of description, but the actual data is in the homomorphically encrypted ciphertext state.

[00238] If the number of people in the metropolitan city of the age 60s or more is to be checked in the data as described above, the count may be carried out by using the bin mask which includes 3 for age and 2 for region.

[00239] In addition, if an average credit rating is to be obtained of a person from Seoul, a relevant slot may be detected by using 1 bin mask for the region, and calculate the average by performing homomorphic addition on data of an encrypted credit rating corresponding to the detected slot. A more detailed conditional statistical operation will be described below.

[00240] A more detailed operation on the statistic calculation described above using the previously described bin mask and expanded bin mask will be described below.

[00241] For convenience of description, a categorical variable may be referred to as a 'bin variable' (or, bin feature), and for convenience of operation, the bin feature will be represented as a positive integer continued from 1. For example, one column of bin variable with three categories may be such that the value of each row is 1 or 2 or 3.

[00242] The system according to the disclosure may provide the statistical

operation on data of which a certain bin variable has a specific value, that is, belonging to a specific category. This is because data in the specific category and the statistical feature may be different because the average, variance and standard deviation operation above is an operation on all data.

[00243] In addition, the statistical operation may be possible on not only the condition on the one bin variable, but also on data having the condition on several bin variables of an arbitrary number.

[00244] The statistical operation as described above is the same as realizing the operation formula as below in the encrypted state.

[00245] Average:

$$\begin{cases} E(x^{(f_0)} | v^{(f_0)} = 1) = \sum_{i=0}^{R-1} x^{(f_0)}(i) \cdot v^{(f_0)}(i) / \sum_{i=0}^{R-1} v^{(f_0)}(i) & (w/v\text{-bit}) \\ E(x^{(f_0)}) = \sum_{i=0}^{R-1} x^{(f_0)}(i) / R & (w/o\ v\text{-bit}) \end{cases}$$

[00246]

[00247] Variance:

$$\begin{cases} Var(x^{(f_0)} | v^{(f_0)} = 1) = E((x^{(f_0)} - E(x^{(f_0)} | v^{(f_0)} = 1))^2 | v^{(f_0)} = 1) & (w/v\text{-bit}) \\ Var(x^{(f_0)}) = E((x^{(f_0)} - E(x^{(f_0)}))^2) & (w/o\ v\text{-bit}) \end{cases}$$

[00248] Correlation Coefficient:

$$\begin{cases} r(x^{(f_0)}, x^{(f_1)} | v^{(f_0)} = 1, v^{(f_1)} = 1) = \frac{E(x^{(f_0)} - E(x^{(f_0)} | cond) | cond) E(x^{(f_1)} - E(x^{(f_1)} | cond) | cond)}{\sqrt{Var(x^{(f_0)} | cond)} \sqrt{Var(x^{(f_1)} | cond)}}, cond = (v^{(f_0)} = 1, v^{(f_1)} = 1) & (w/v\text{-bit}) \\ r(x^{(f_0)}, x^{(f_1)}) = \frac{E(x^{(f_0)} - E(x^{(f_0)})) E(x^{(f_1)} - E(x^{(f_1)}))}{\sqrt{Var(x^{(f_0)})} \sqrt{Var(x^{(f_1)})}} & (w/o\ v\text{-bit}) \end{cases}$$

[00249]

[00250]

[00251] It may be assumed that index $f_0, f_1, \dots, f_{m-1}$ of the bin features are present, and integer $c_0, c_1, \dots, c_{m-1}$ is present. The operation below may perform the

statistical operation with the condition on $x^{(f_0)}=c_0, x^{(f_1)}=c_1, \dots, x^{(f_{m-1})}=c_{m-1}$ bin.

[00252] [Equation 9]

$$cond = \begin{cases} (x^{(f_0)}=c_0, x^{(f_1)}=c_1, \dots, x^{(f_{m-1})}=c_{m-1}, v^{(f_0)}=1, v^{(f_1)}=1, \dots, v^{(f_{m-1})}=1, v^{(f_{i-1})}=1) & (w/v-bit) \\ (x^{(f_0)}=c_0, x^{(f_1)}=c_1, \dots, x^{(f_{m-1})}=c_{m-1}) & (w/ov-bit) \end{cases}$$

[00253]

[00254] Bin Count: function counting a number of possible cases satisfying

the condition in $\text{sum}(x^{(f_i)}|cond)$ data table.

[00255] Average: total of $x^{(f_i)}$ column values satisfying $\text{sum}(x^{(f_i)}|cond)$ condition.

[00256] Bin Average: $E(x^{(f_i)}|cond) = \text{sum}(x^{(f_i)}|cond) / \text{cnt}(x|cond)$

[00257] Bin Variance: $Var(x^{(f_i)}|cond) = E((x^{(f_i)} - E(x^{(f_i)}|cond))^2 | cond)$

[00258] A more detailed statistical operation method will be described with reference to FIGS. 25 to 31.

[00259]

[00260] A comparison operation between the bin variable column and the specific integer value is necessary for the process described above.

[00261] However, because the comparison operation using the homomorphic operation involves significant cost, the comparison results may be stored in advance in the bin mask and utilized in the operation.

[00262] A certain bin feature f_0 may include s_0 as a maximum bin variable. That is, each row of one column $\overrightarrow{x^{(f_0)}}$ of the data may include one value from among $\{1, 2, \dots, s_0\}$. The bin mask generated at this time may be encrypted columns of a total s_0 number, and may be represented as $b_i^{(f_0)}$ ($i=1, 2, \dots, s_0$). Here, $b_i^{(f_0)}$ is one encrypted column, and each row may be shown as 1 or 0 if the row corresponding to bin feature f_0 is i . Further, $b_{i,j}^{(f_0)}$ may refer to j -th block of $b_i^{(f_0)}$.

[00263] The operation of generating the bin mask used in the statistical operation will be described in greater detail below.

[00264] The bin mask may be generated at the encryption step, and may be generated even after the encryption step. First, the operation of generating the bin mask at the encryption step will be described below with reference to FIG. 9.

[00265]

[00266] FIGS. 9 and 10 are diagrams illustrating a process of generating a bin mask by using plaintext according to an embodiment of the disclosure. Specifically, the diagrams are for describing the process of generating the bin mask taking into reference data in the plaintext state when proceeding with encryption from the data owner side. Specifically, FIG. 9 is when a v-bit table is used, and FIG. 10 is when a v-bit table is not used.

[00267] Referring to FIGS. 9 and 10, an index of bin features for forming a bin mask may be vector $\vec{f} = (f_0, f_1, \dots, f_{m-1})$, and the maximum bin value of each bin feature may be represented as $\vec{s} = (s_0, s_1, \dots, s_{m-1})$.

[00268] The data table in the plaintext state may be represented as $\mathbf{x}$, and the v-bit table may be represented as $\mathbf{v}$. The output value may be $\mathbf{b} = \{b^{(f_i)} | i=0, 1, \dots, m-1\}$.

[00269]

[00270] The generating operation of the bin mask after the encryption operation will be described below.

[00271] The generating of the bin mask after the encryption operation may be used when data in the plaintext state cannot be approached, and when generating the bin mask using the homomorphic operation in the encrypted state.

[00272] First, a realization of an integer comparing operation as below is

required in the encrypted stated for the bin mask generation.

[00273] [Equation 10]

$$[00274] \quad (a \equiv b) = \begin{cases} 1 & \text{if } a=b \\ 0 & \text{if } a \neq b \end{cases}$$

[00275] This operation may be represented as $(a \equiv b) = \delta(a-b)$ with respect to a function as with Equation 11 below.

[00276] [Equation 11]

$$[00277] \quad \delta(x) = \begin{cases} 1 & \text{if } x=0 \\ 0 & \text{if } x \neq 0 \end{cases}$$

[00278]

[00279] Because the categorical variable includes all integer values, it is to satisfy $\delta(x)$ function with respect to the input of the integer value. Because addition and multiplication are provided in the homomorphic encryption system, the function is to be satisfied within the integer range by approximating the $\delta(x)$ function with any polynomial.

[00280] At this time, a sinc function as below may be used as an approximation function

[00281] [Equation 12]

$$[00282] \quad \text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$$

[00283] The above may include a value such as $\delta(x)$ from all integer values, and a polynomial approximation may be possible. However, when approximating in the wide range, the amount in calculation may increase as the degree of an approximation formula becomes higher. In this case, approximation in a wide range may be possible with only the approximation of a narrow range by using a multiple angle formula of a

trigonometric function.

[00284] First, using the $\sin 2\phi = 2\sin\phi\cos\phi$ equation, the sinc function may be changed like in Equation 13 as below.

[00285] [Equation 13]

$$\begin{aligned}\text{sinc}(x) &= \frac{\sin(\pi x)}{\pi x} = \frac{2}{\pi x} \sin\left(\frac{\pi x}{2}\right) \cos\left(\frac{\pi x}{2}\right) \\ &= \text{sinc}\left(\frac{x}{2^d}\right) \cos\left(\frac{x}{2^d}\right) \cos\left(\frac{x}{2^{d-1}}\right) \dots \cos\left(\frac{x}{2}\right)\end{aligned}$$

[00286]

[00287] If $\cos\left(\frac{x}{2^d}\right)$ value is known, because the value of $\cos\left(\frac{x}{2^{d-1}}\right), \dots, \cos\left(\frac{x}{2}\right)$ can all be known by using the $\cos 2\phi = 2\cos^2\phi - 1$ formula,

the $\text{sinc}(x)$ value may be obtained if the values of $\text{sinc}\left(\frac{x}{2^d}\right)$ and $\cos\left(\frac{x}{2^d}\right)$ are known. Accordingly, the $\text{sinc}(x)$ value may be known in $[-\mu, \mu]$ with only

approximating $\text{sinc}(x)$ and $\cos(x)$ in $\left[-\frac{\mu}{2^d}, \frac{\mu}{2^d}\right]$ range.

[00288] A Chevyshev approximation may be used as below for the approximation of the $f(x) = \text{sinc}(x)$ or $f(x) = \cos(x)$ function. However, other approximation methods may be used at realization.

[00289] [Equation 14]

[00290]

$$f(x) \approx \frac{1}{2} c_0 + \sum_{j=1}^K c_j T_j(x)$$

[00291] Here, c_j is the same as in Equation 15.

[00292] [Equation 15]

[00293]
$$c_j = \frac{2}{K+1} \sum_{k=1}^{K+1} \left[\cos \left\{ \frac{\pi(2k-1)}{2(K+1)} \right\} \right] \cos \left\{ \frac{\pi j(2k-1)}{2(K+1)} \right\}$$

[00294] Here, $T_j(x)$ is a J -th Chevyshev polynomial.

[00295] The polynomial generated according to the approximation method described above may be a K -th polynomial.

[00296]

[00297] When using the method as described above, approximation may be carried out with a small error in $[-1, 1]$ range, and if K is an even number, the approximation formula may include a value of 1 exactly when $x=0$.

[00298] If there is an error present, because the value of the bin mask may not include a value of 1 exactly as the error is continuously amplified when calculating $\text{sinc}(x)$ using the multiangle formula, K may be set as an even number.

[00299] FIG. 11 is a diagram illustrating an approximation algorithm according to an embodiment of the disclosure. Specifically, the DiscreteEqualZero algorithm in FIG. 11 represents the approximation algorithm of $\text{sinc}(x)$ function using the above-described method.

[00300] Referring to FIG. 11, each slot may receive ciphertext $\mathbf{x}$ which is an integer value as an input, and return the ciphertext of which the slot, with the value of the corresponding slot being 0, is 1 and the rest of the slot is 0. By using the approximation formula above, it may be assumed that coefficient $a_i, b_i (i=0, 1, \dots, K)$

of $\text{sinc}(x) \approx \sum_{i=0}^K a_i x^i$, $\cos(\pi x) \approx \sum_{i=0}^K b_i x^i$ is calculated in advance.

[00301] Each slot of $\mathbf{x}$ with respect to μ of the input may be a value in $[-\mu, \mu]$ range, and K may be an even number as a degree of approximation polynomial. M of the process may be the number of slots in one ciphertext.

[00302] At this time, the multiplication operation may be $O(K+3 \lceil \log_2 \mu \rceil)$ time.

[00303] The bin mask generating operation which uses the integer comparing operation by using the method described above will be described with reference to FIG. 12.

[00304] FIG. 12 is a diagram illustrating a bin mask generating operation using a homomorphic ciphertext according to an embodiment of the disclosure.

[00305] Referring to FIG. 12, the index of the bin features creating the bin mask may be vector $\vec{f}=(f_0, f_1, \dots, f_{m-1})$, and the maximum bin value of each bin feature may be represented as $\vec{s}=(s_0, s_1, \dots, s_{m-1})$.

[00306] Encrypted data table X, and encrypted v-bit table V may be received as input.

[00307] The output value may be $b = \{b^{(f_i)} | i=0, 1, \dots, m-1\}$. If all slot values are valid in the encrypted state, one multiplication operation may be reduced in rows 8 and 9 of FIG. 12.

[00308] A detailed method of calculating the bin count by using the generated bin mask will be described below.

[00309] FIG. 13 is a diagram illustrating an operation of a bin count operation according to an embodiment of the disclosure.

[00310] Here, the bin count operation may be a function which counts the number of rows satisfying the condition on the several bin features.

[00311] When receiving vector $\vec{c}=(c_0, c_1, \dots, c_{m-1})$ formed of vector $\vec{f}=(f_0, f_1, \dots, f_{m-1})$ and integer value of the bin feature as an input, the bin count operation may include counting the number of valid data which satisfies m number of conditions of f_0 -th feature value being c_0 , ... f_{m-1} -th feature value being c_{m-1} .

[00312] When using the previously generated bin mask, because v-bit was taken into reference in the previous bin mask process, a separate consideration on v-bit is not necessary.

[00313] If all conditions are satisfied when all the bin masks corresponding to the condition are multiplied row-wise, the value may be 1 and if not 0.

[00314] Accordingly, the bin count operation may add the value of all slots of the multiplication result.

[00315] In FIG. 13, as input $b = \{b^{(f_i)} | i=0,1,\dots,m-1\}$ of the bin count operation, it is a set of bin masks which is the output value from the previous bin count generating process.

[00316] Further, when n_b is the number of blocks in one column, m is the number of conditions, and M is the number of slots in one ciphertext, the process above may perform $O(n_b * m)$ number of multiplications and $O(\log 2M)$ rotation operation.

[00317]

[00318] A modification of the above-described operation of the bin count operation will be described below.

[00319] First, the modified bin count method (hereinbelow, referred to as a large bin count) may obtain all number of possible cases formed by the several bin features of a certain data and represent the result in table form.

[00320] The specific operation will be described with reference to FIG. 14.

[00321] FIG. 14 is a diagram illustrating original data and a target of a large bin count operation according to an embodiment of the disclosure.

[00322] Referring to FIG. 14, the result table of the maximum bin variable seeking five bin features A, B, C, D, E is shown.

[00323] If a large bin count is performed on m number of bin features having $s_0, s_1, \dots, s_{m-1}$ with the maximum bin value respectively, a result table representing all $\prod_{i=0}^{m-1} s_i$ counts of number of possible cases may be obtained.

[00324] The bin count method may use the bin mask which was previously formed as described above. Bin mask $b_i^{(f_0)}$ may be an encrypted column which represents whether the value of each row of a certain bin feature f_0 matches with i (i is a positive integer) as 1 or 0. (If it is a match with i may be shown as 1, and if not as 0.)

[00325] As illustrated in FIG. 14, a method of counting the number of possible cases when all five features include a value of 1 is as described below. The method will be described with reference to FIG. 15.

[00326] FIG. 15 is a diagram illustrating a method of calculating a number of a specific number of cases by using a bin mask.

[00327] Referring to FIG. 15, if bin masks $b_1^{(A)}, b_1^{(B)}, b_1^{(C)}, b_1^{(D)}, b_1^{(E)}$ are all multiplied, the value of 1 may remain only in s row when all features are 1, and the rest may all be 0.

[00328] Accordingly, the number of possible cases sought may be obtained by adding the value of all rows of the multiplication result.

[00329] However, only the number of one possible case from among all combinations may be obtained through the process of multiplying and adding the bin masks as described above.

[00330] Accordingly, when bin features $f_0, f_1, \dots, f_{m-1}$ of m number include $s_0, s_1, \dots, s_{m-1}$ as maximum bin value, the process above is to be repeated by $\prod_{i=0}^{m-1} s_i$

times in order to obtain the number of possible cases of all combinations formed by the features.

[00331] The data table may include rows of n number, and when each ciphertext includes slots of M number, each row may include blocks of $\lceil n/M \rceil$

number. Accordingly, multiplication of a total $\lceil n/M \rceil * (m-1) * \prod_{i=0}^{m-1} s_i$ number is required.

[00332] The number of bin features seeking the number of possible cases, or the number of multiplication operations necessary as each of the maximum bin value increases may increase.

[00333] Accordingly, a method for calculating the number of possible cases with multiplication of a small number will be described below.

[00334] In order to perform the object as described above, the power bin mask may be used in place of the bin mask.

[00335] Each row of the power bin mask may include a value of $2^{(i-1) \times \Delta + \delta}$ when the value of the corresponding bin feature is i . That is, a margin of about Δ may be provided for each bin value.

[00336] FIG. 16 is a diagram illustrating an operation of a bin count operation using a power bin mask according to an embodiment of the disclosure.

[00337] Referring to FIG. 16, described is an example of forming the power bin mask 1620 by setting as $\delta=0, \Delta=3$, and using feature A 1610. Each row is represented in binary notation.

[00338] Based on the power bin mask 1620 being generated as described above, $p^{(A)}, b_1^{(B)}, b_1^{(C)}, b_1^{(D)}, b_1^{(E)}$ 1630 may be multiplied as in the previous method.

[00339] The value of each row in the multiplication result may be 0 when

features B, C, D and E are all not 1. Then, the value of all rows may be added. If it can be ensured that the number of rows being added at this time is less than 2^{Δ} , the addition result may be respectively stored in an area by about Δ bit.

[00340] The example in FIG. 16 is a result of having added only the provided six columns 1640. The value in the lowest 3 bit at this time shows the number of possible cases in which all features are 1, and the value stored in the highest 3 bit shows the number of possible cases in which feature A is 5 and the rest is 1.

[00341] The result value may include both B, C, D and E being 1, and five counts of number of possible cases when $A = 1 \sim 5$. Accordingly, although the method may include performing the multiplication operation of the same number as the previous bin count method, it may be equivalent to obtaining a number of possible cases which is 5 times more.

[00342] Accordingly, bin features $f_0, f_1, \dots, f_{m-1}$ of m number may include $s_0, s_1, \dots, s_{m-1}$ as the maximum bin value, and when trying to obtain the number of possible cases of all combinations thereof with f_0 , the number of multiplication times may be reduced by $1/s_0$ times in theory when a bin mask expanded with f_0 is generated.

[00343] In order to apply the power bin mask method as described above, each row of the bin mask is to be represented as 0 or 2^{Δ} (here, Δ is a positive integer) rather than 0 or 1 considering an error term of the homomorphic encryption method.

[00344] When bin mask is multiplied by $m-1$ number, the value of each row is increased by up to a maximum $2^{(m-1)\Delta}$, and the value is not to exceed a modulus bit of the homomorphic ciphertext.

[00345] To this end, a process of representing bin features of $m-1$ number as new bin features of k number may be included ($m-1 > k$).

[00346] This is because if a bin mask is formed and multiplied with one another in the method described above with respect to the new bin features of k number, the maximum value of each row may be reduced by $2^{k\Lambda}$.

[00347] The whole bin count operation which considers the error term as described above is the same as in FIG. 17.

[00348]

[00349] FIG. 17 is a diagram illustrating a bin count operation considering an error term according to an embodiment of the disclosure.

[00350] Referring to FIG. 17, bin features 1740 and 1750 of four B, C, D and E may be formed as two new bin features and a new big bin mask may be formed. When a big bin mask as described above is formed, the big bin mask may be multiplied to the previously generated power big mask 1730 to obtain the number of possible cases.

[00351] In summarizing the operation of the bin count operation as described above, first the power bin mask and the big bin mask may be generated proactively. The bin mask process as described above may be performed in the encryption process. Alternatively, it may be performed after the encryption process.

[00352] When the power bin mask and the big bin mask are generated, the multiplication operation between the relevant masks may be performed. Then, a decryption process for checking the operation result may be performed.

[00353] Specifically, an expanded bin mask may be generated with one feature (f_0) for the large bin count operation.

[00354] Then, a big bin mask of k number may be generated with the rest of $(f_1, f_2, \dots, f_{m-1})$ of $m-1$ number ($k < m-1$). This is to represent columns of $m-1$ number as columns of k number which is a smaller number, and a process of forming a bin mask of a new column. For convenience of description, a case of $k=2$ will be

described below. The big bin mask generated as described above may be referred to as big bin mask 1 and big bin mask 2, respectively.

[00355] At this time, the generated expanded bin mask, big bin mask 1, and big bin mask 2 may be represented as $\mathbf{P}$, $\mathbf{q}^{(i)}$, $\mathbf{r}^{(i)}$ ($i=1,2,\dots,Q$), respectively. ($\mathbf{P}$, $\mathbf{q}^{(i)}$, $\mathbf{r}^{(i)}$ ($i=1,2,\dots,Q$) may be equivalent to one column encrypted respectively. Accordingly, it may be comprised of ciphertext by a block number ($\lceil n/M \rceil$) for one column, respectively.)

[00356] [Equation 16]

[00357] $\mathbf{q}=\{ \mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \dots, \mathbf{q}^{(Q)} \}$, $\mathbf{r}=\{ \mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots, \mathbf{r}^{(Q)} \}$

[00358] $\mathbf{P}_{k^{(i,j)}}$, $\mathbf{q}_{k^{(i,j)}}(j)$, $\mathbf{r}_{k^{(i,j)}}(j)$: a value relevant to $j+1$ -th slot of $k+1$ -th ciphertext in each mask ($i=0,1,\dots,m-1$, $k=0,1,\dots, \lceil n/M \rceil -1$, $j=0,1,\dots,M-1$)

[00359]

[00360] FIG. 18 is a diagram illustrating a generation operation of a power bin mask according to an embodiment of the disclosure.

[00361] Referring to FIG. 18, the expanded bin mask may be represented such that the value of each slot is $2^{(i-1) \times \Delta + \delta}$, $i(\in [1,50])$.

[00362] Because an error is generated at a lower bit in a specific homomorphic encryption system, an offset of lower δ bit is provided to each slot value of the power bin mask.

[00363] x, v, f_0 of input may be an index of the bin feature for forming a data table, v-bit table in plaintext state, and power bin mask, respectively.

[00364] Further, n may be a number of data rows, and M may be a number of slots of one ciphertext.

[00365]

[00366] FIG. 19 is a diagram illustrating a generation operation of a power bin mask according to another embodiment of the disclosure.

[00367] Referring to FIG. 19, assuming that bin variables of $m-1$ number excluding the one used to form the power bin mask are represented as two new bin features, described is a process of generating a mask on the new bin features.

[00368] A typical bin mask represented whether it is relevant to a specific value as 0 or 1, but it may be represented as 0 or 2^A in this case.

[00369] If a feature index of $m-1$ number is $f_1, f_2, \dots, f_{m-1}$ respectively, and the maximum bin value of each feature is $s_1, s_2, \dots, s_{m-1}$, the new two variables may

each include $Q = \lceil \sqrt{\prod_{i=1}^{m-1} s_i} \rceil$ as the maximum bin variable. Here, x, v may be a data table in plaintext state and v -bit table in plaintext state, respectively.

[00370] As a result of this process, bin masks on the two new assumed columns may be generated. That is, it may be equivalent to columns of 2^*Q number, of which the number of rows is n , being generated.

[00371] This process is so that the value stored in each slot of the multiplication process of the large bin count is not greater than or equal to the modulus bit.

[00372] As described above, when the large bin count and the big bin count are prepared, the multiplication operation may be performed.

[00373] Specifically, all $p_k \cdot q_k^{(i)} \cdot r_k^{(j)}$, ($i, j = 1, 2, \dots, Q$, $k = 0, 1, \dots, \lceil n/M \rceil - 1$) may be calculated. A GPU may be used in the homomorphic multiplication operation of this process, and each GPU may perform the multiplication operation on a block basis. The specific operation will be described below with reference to FIG. 20.

[00374] FIG. 20 is a diagram illustrating an operation of a multiplication

operation between a plurality of bin masks according to an embodiment of the disclosure. Further, FIG. 21 is a diagram illustrating an operation of a multiplication operation using a plurality of GPUs.

[00375] Referring to FIG. 20, each GPU 2010, 2020, 2030 and 2040 may be operated in parallel. Accordingly, if the number of GPUs is N_g number, blocks of a total of N_g number may be simultaneously processed for every one block. The GPUs 2010, 2020, 2030 and 2040 may at first be allocated with block jobs of N_g number, and after completing the job, process the job in a manner of receiving allocation of the next block job of N_g number in order. Then, the number of ciphertexts loading in the GPU all at once may be limited according to the memory capacity of the GPU.

[00376] After processing one block, a result ciphertext may be generated by Q^2 number.

[00377] When the above is stored with respect to all blocks, it may become $\lceil n/M \rceil * Q^2$ number and this may require greater storage space as the block number increases. Accordingly, if the number of blocks is greater than a number (N_g) of GPU, the result ciphertext of the previously processed block may be called and stored after adding with the new result ciphertexts. Accordingly, the number of ciphertexts which may be stored may be limited to a maximum $N_g * Q^2$.

[00378] When the GPU is able to load one expanded bin mask, big bin mask 1 of c number, and big bin mask 2 of c number all at once in the operation process, the algorithm may be as described in FIG. 21.

[00379] In FIG. 21, c is a certain positive integer, and may be determined according to a size of the memory of the GPU being used. The mask columns (P, Q, R) generated in the previous process, and the number of GPUs for performing the operation may be received as input.

[00380] Because the mask generated through the process as described above is in an encrypted state, utilizing the data as is may be difficult. Accordingly, the decryption operation after the operation of multiplication operation will be described below.

[00381] FIG. 22 is a diagram illustrating a decryption operation after multiplication operation according to an embodiment of the disclosure.

[00382] Referring to FIG. 22, first, based on a review of the elements of $\text{mres} = \{\text{res}_k^{(ij)} | k=0, 1 \dots N_g-1, i, j=1, 2 \dots Q\}$ which are output values of a previous operation, the number of possible cases of (i, j) with respect to value i of BigBin1, and value j of BigBin2 may be a total Q^2 .

[00383] Because BigBin1, BigBin2 represent $m-1$ number features as two features, (i, j) pair may be mapped as one possible case from among the combination of bin features of $m-1$ number. There may be N_g number of ciphertexts with the value of BigBin1 being i , and the value of BigBin2 being j .

[00384] Accordingly, after decryption of all ciphertexts of N_g number, by cutting the value of each slot to a bit unit and adding only the values of Δ bits of $(2 \cdot \Lambda + l \cdot \Delta + \delta) \approx (2 \cdot \Lambda + (l-1) \cdot \Delta + \delta)$, the number of possible cases in which the bin feature f_0 value which formed the expanded bin mask is 1 may be obtained.

[00385] As illustrated above, with the input value of the relevant algorithm, the result of the multiplication process of the previous bin mask is required. With the result of this process, a table of $Q^2 \times s_0$ size containing number of all possible cases may be obtained.

[00386] FIG. 23 is a diagram illustrating a data structure of a bin mask according to an embodiment of the disclosure. Specifically, FIG. 23 is a diagram showing an area taken up by the values of each slot when a power bin mask 2310 and

big bin masks 2320 and 2330 are generated.

[00387] Referring to FIG. 23, because the maximum bin value of feature f_0 is s_0 , the power bin mask may include one value from among $2^{(\Delta \cdot 0 + \delta)}, 2^{(\Delta \cdot 1 + \delta)}, \dots, 2^{(\Delta \cdot (s_0 - 1) + \delta)}$. Then, the big bin masks $q_{k^{(i)}}, r_{k^{(i)}}$ may include a value of 2^Δ or 0. Accordingly, it may be equivalent to only one bit of Δ -th bit being represented as 0 or 1.

[00388] FIG. 24 is a diagram illustrating a data structure of a multiplication operation result according to an embodiment of the disclosure. Specifically, FIG. 24 shows a use of modulus bit of an arbitrary slot obtained as a result of a multiplication operation process.

[00389] Referring to FIG. 24, by adding the results processed in the same GPU, the areas of about Δ bit as in the drawing below may all be shown as representing the number of possible cases where the value of the bin feature which generated power bin mask is l . Because the area representing the bin feature value being $l+1$ is not to be intruded because of the value increasing, the maximum number of blocks $\lceil \lceil n/M \rceil / N_g \rceil$ processed by one GPU is not to exceed 2^Δ .

[00390] If the description illustrated in FIG. 24 is represented in a formula, one slot of the power bin mask may be $2^{(i-1) \times \Delta + \delta} + e$, $i \in [1, s_0]$, and the big bin masks may be $a \cdot 2^\Delta + e$, $(a \in \{0, 1\})$.

[00391] What is to be obtained through this operation is whether it is relevant to a specific bin value, and may be identified through a value from lower $(2\Delta + s_0 \cdot \Delta + \delta)$ -th bit to $(2\Delta + s_0 \cdot \Delta + \delta)$ bit of an arbitrary slot of a multiplication result.

[00392] The multiplication result of an arbitrary slot may be represented as in Equation 17 below.

[00393] [Equation 17]

$$\{2^{(i-1) \times \Delta + \delta} + e\} \cdot \{a_1 \cdot 2^\Delta + e\} \cdot \{a_2 \cdot 2^\Delta + e\}$$

[00395]

$$\approx a_1 a_2 \cdot 2^{((i-1) \cdot \Delta + \delta + 2\Delta)} + \{2^{2\Delta} + 2 \cdot 2^{((i-1) \cdot \Delta + \delta + 2\Delta)}\} \cdot e \quad (a_1 a_2 \in \{0, 1\})$$

[00396] At this time, the lower bit of the value desired by the error of the multiplication result is not to be exceeded, and an upper bit of the value is not to exceed the modulus bit. That is, the error term $(2^{2\Delta} + 2^{((s_0-1) \cdot \Delta + \delta + \Delta) + 1}) \cdot e$ of the multiplication result may not intrude $2^{2\Delta + \delta}$, and the maximum bit $(2\Delta + (s_0 - 1) \cdot \Delta + \delta)$ may not be greater than the modulus bit used in the homomorphic encryption system.

[00397] Considering that the result generated in another block is added to thereto after the multiplication, the equation is to be adjusted to have a margin of about Δ bit in the condition above. The result is as follows. It may be assumed that b_m is the modulus bit, and a log upper bound of the error term is $b_e (|e| < \text{pow}(2, b_e), b_e < 0)$. Equation 18 below may be an equation on the error term, and Equation 19 may be an equation on the modulus bit.

[00398] [Equation 18]

$$(2^{2\Delta} + 2^{((s_0-1) \cdot \Delta + \delta + \Delta) + 1}) \cdot e < 2^{2\Delta + \delta - \Delta}$$

[00400] [Equation 19]

$$(2\Delta + s_0 \cdot \Delta + \delta) < b_m$$

[00402] Accordingly, Δ, δ, Δ may be set to satisfy the condition of $\lceil \lceil n/M \rceil / N_g \rceil < 2^\Delta$ and two inequations above.

[00403] Although the big bin mask has been formed by representing features of m-1 number as two columns in the above-described process, it may be possible to also represent as columns of arbitrary natural numbers of k-number.

[00404] When representing as columns of k-number, it is to be set as

$Q = \lceil \sqrt[k]{\prod_{i=1}^{m-1} s_i} \rceil$, and it is to be divided repeatedly with Q by k times in the big bin mask generating process, and the rest may be set to a bin value of a new column.

[00405] In this case, the result of multiplication may be

$$\begin{aligned} & \{2^{(i-1) \cdot \Delta + \delta} + e\} \cdot \prod_{j=1}^k a_j \cdot 2^{\Lambda} + e \\ \approx & a \cdot 2^{((i-1) \cdot \Delta + \delta + k\Lambda)} + \{2^{k\Lambda} + k \cdot 2^{((i-1) \cdot \Delta + \delta + (k-1)\Lambda)}\} \cdot e \\ & (a, a_j \in \{0, 1\}) \end{aligned}$$

[00406] and a limiting formula on the error term and modulus bit above may be described as below.

[00407] [Equation 20]

$$\text{[00408]} \quad (2^{k\Lambda} + k2^{((s_0-1) \cdot \Delta + \delta + (k-1)\Lambda)}) \cdot e < 2^{2\Lambda + \delta - \Delta}$$

[00409] In the large bin count process as described above, the homomorphic multiplication operation of $\lceil n/M \rceil (Q^2 + Q)$ number may be used.

[00410]

[00411] FIG. 25 is a diagram illustrating a comparison operation according to the disclosure.

[00412] In order to calculate the statistic calculation, a comparison of values in the homomorphic ciphertext may be necessary. That is, it may be necessary to check whether the ciphertext value in the homomorphic ciphertext corresponds to a specific variable value. FIG. 25 shows a comparison algorithm for the above.

[00413] Referring to FIG. 25, two variables may be received, and a comparison result may be calculated through the operation by the two variables. In the homomorphic ciphertext state, a comparison result may be calculated by using the approximation function on the sinc function as described above.

[00414]

[00415] FIGS. 26 to 28 are diagrams illustrating various statistic calculation methods according to an embodiment of the disclosure.

[00416] Specifically, FIG. 26 shows an algorithm 2600 on an average calculation method.

[00417] Referring to FIG. 26, when an index vector $\vec{f}=(f_0, f_1, \dots, f_{m-1})$ of the bin feature and a vector $\vec{c}=(c_0, c_1, \dots, c_{m-1})$ representing a conditional bin value are present, the average value of feature f_i of rows satisfying m number conditions of f_0 -th feature value being c_0 , ... and f_{m-1} -th feature value being c_{m-1} may be calculated. This may be equivalent to applying the average (Avg) process described previously on a column including f_i column as data, and multiplication of bin masks satisfying the condition with v-bit. To this end, the process of determining data column $X^{(f_i)}$ and v-bit column $V^{(f_i)}$ on a temporary feature f^* may be performed.

[00418] The whole process is as follows. X, V may respectively be the encrypted data table and the v-bit table, and b may be a set of BinMask. $\vec{f}, \vec{c}$ may represent the condition on the bin variable, f_i may represent the index of the column seeking the average, and f^* may represent the index of the column temporarily generated in the process of seeking the average. k may be an iteration number of ApproxIn in an average operation.

[00419] The process above may be equivalent to multiplication of $O(m)$ number being added to the Average process. Accordingly, multiplication of $O(n_b + 2k + m)$ and rotation process of $O(21\log 2(M))$ number may be required. (n_b is the block number of one column, k is the iteration number in an inverse process of the average operation, and M is the number of slots per one ciphertext.)

[00420]

[00421] Further, FIG. 27 is a diagram illustrating an algorithm 2700 on a method of calculating variance.

[00422] Referring to FIG. 27, like the bin average process above, this time a variance rather than the average may be obtained. Likewise, the variance operation described previously may be applied by generating a temporary data column $X^{(f_*)}$ and v-bit column $V^{(f_*)}$. The process is as described below. X, Y may be the data table and the v-bit table respectively, and b may be the BinMask set. $\vec{f}, \vec{c}$ may represent the condition on the bin variable, f_i may represent the index of the column seeking the average, and f^* may represent the index of the column which is temporarily generated in the process seeking the average. k may represent the iteration number of ApproxIn in the average operation.

[00423] The process above may be equivalent to the multiplication process of about $O(m)$ being added to the previous variance process. Accordingly, multiplication of $O(3n_b + 2k + m)$ and rotation operation of $O(3\log_2(M))$ may be necessary. (n_b is the block number of one column, k is the iteration number in an inverse process of the average operation, and M is the number of slots per one ciphertext.)

[00424]

[00425] FIG. 28 is a diagram illustrating an algorithm 2800 on a method of calculating a correlation coefficient.

[00426] Pearson correlation coefficient between the two features f_0, f_1 of the data table may be operated. At this time, operation may be performed only on rows in which the values of the two features are all valid taking into reference the v-bit table.

$$r_{XY} = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2} \sqrt{\sum(Y - \bar{Y})^2}}$$

The correlation coefficient formula may be used on the two variables X, Y. The algorithm seeking the correlation coefficient of the two features f_0, f_1 in the encrypted data table and the encrypted v-bit table in the homomorphic encryption method is as follows. The iteration number of `ApproxIn` in the operation process may be k2, and the iteration number of `ApproxSqrtInv` may be k1. Likewise, when the value of all slots is ensured to be valid, it may be possible in even operations not using v-bit.

[00427]

[00428] The difficult part in the above-described job is to find a reciprocal number of a number associated with the job. The reason finding the reciprocal number is difficult is that a range of values required in a reverse calculation is to be set, a parameter is to be set so that the result is not diverged in the range, and that approximation algorithm is mainly comprised of repeat algorithm. Accordingly, for the accuracy of the result, the number of repeating times is to be increased, but because calculation costs increase as the number of repeating times increases, an appropriate number of repeating times is to be performed. Because homomorphic ciphertext includes errors, after a certain number of operation times, a rebooting operation is to be performed.

[00429]

[00430]

[00431] FIG. 29 is a diagram illustrating an operation of calculating a maximum value in a slot according to the disclosure.

[00432] Referring to FIG. 29, a relevant algorithm 2900 may store the value of

one slot in the homomorphic ciphertext, sequentially compare the stored value with another slot value, and calculate the maximum value. The comparison operation may employ the comparison algorithm illustrated above.

[00433]

[00434] FIG. 30 is a diagram illustrating an operation of calculating a maximum value in several columns in a plurality of blocks.

[00435] Referring to FIG. 30, a relevant algorithm 3000 may first calculate a maximum value with the highest value for each block by using the algorithm as in FIG. 29, and calculate the maximum value in the plurality of blocks through a comparison between the calculated maximum values.

[00436] In the above, only the operation of calculating the maximum value has been described, but it is also possible to calculate a minimum value through an operation of calculating a small value in the comparison process.

[00437]

[00438] FIG. 31 is a diagram illustrating a method of calculating a value of a specific order according to an embodiment of the disclosure.

[00439] Referring to FIG. 31, a percentile algorithm 3100 is similar to the method of arranging data in an ascending order. First, the arrangement process may be performed, and a value corresponding to the percentile requested thereto may be calculated.

[00440]

[00441] FIG. 32 is a flowchart illustrating a ciphertext processing method according to an embodiment of the disclosure.

[00442] Referring to FIG. 32, first the statistic calculation command on the plurality of homomorphic ciphertext may be received (S3210). The statistic calculation

command as described above may include calculating the number of variables having a specific value, an average of values satisfying the specific condition, variance, and the like.

[00443] Then, the plurality of homomorphic ciphertext may be stored with the encrypted state of the plurality of variable data.

[00444] Then, the bin mask having different variable data classified on each of the homomorphic ciphertext may be generated (S3220). The generating of the bin mask as described above may be generated in the generation process of the homomorphic ciphertext as previously described, and may also be generated in the homomorphic ciphertext state. Further, the bin mask may be a bin mask including only one variable data per slot, and may be the expanded bin mask including whether the plurality of variable values is present, the power bin mask previously described, the big bin mask, or the like.

[00445] Then, number data corresponding to the variable combination may be generated by using the bin mask (S3230). Specifically, a count value matching a specific condition may be calculated by using the multiplication of the generated bin mask.

[00446] Then, the calculated number data may be output. The output as described above may be performed in the encrypted state, the process of decrypting the relevant data may be performed, and may be output as the decrypted result.

[00447] Accordingly, the encryption processing method according to an example embodiment may perform an efficient statistical operation on the homomorphic ciphertext. The ciphertext processing method as in FIG. 32 may be executed on the electronic device which includes the configuration of FIG. 2, and may also be executed on electronic devices which include other configurations in addition

thereto.

[00448] In addition, the ciphertext processing method as described above may be realized with a program including an algorithm executable in a computer, and the above-described program may be stored in a non-transitory computer readable medium and provided.

[00449] The non-transitory computer readable medium may refer to a medium that stores data semi-permanently rather than storing data for a very short time, such as a register, a cache, a memory, or the like, and is readable by a device. Specifically, programs for performing the various methods described above may be stored in the non-transitory computer readable medium such as, for example, and without limitation, a compact disc (CD), a digital versatile disc (DVD), a hard disc, a Blu-ray disc, a USB, a memory card, a ROM, and the like, and provided.

[00450] In addition, while the disclosure has been shown and described with reference to the example embodiments thereof, the disclosure is not limited to the specific example embodiments described above, and various modifications may be made therein by those skilled in the art to which this disclosure pertains without departing from the spirit and scope of the disclosure, and such modifications shall not be understood as separate from the technical concept or outlook of the present disclosure.

WHAT IS CLAIMED IS:

1. An electronic device, comprising:

a memory configured to store at least one instruction, and store a plurality of homomorphic ciphertexts, each of the plurality of homomorphic ciphertexts storing a plurality of variable data in an encrypted state; and

a processor configured to execute the at least one instruction,

wherein the processor is configured to generate, by executing the at least one instruction, a bin mask corresponding to a variable combination and having different variable data of the plurality of variable data classified for each of the plurality of homomorphic ciphertexts, and to generate number data corresponding to the variable combination by using the bin mask and the stored plurality of variable data based on a received instruction to perform an operation instruction on the plurality of homomorphic ciphertexts being received.

2. The electronic device of claim 1, wherein each of the plurality of homomorphic ciphertexts comprises a plurality of slots, and

each of the plurality of slots comprises one variable data.

3. The electronic device of claim 1, wherein the bin mask comprises a plurality of slots,

wherein each of the plurality of slots comprises data on whether one variable value is present,

wherein the processor is configured to generate a bin mask for each of the plurality

of variable data comprised in each of the plurality of homomorphic ciphertexts, select bin masks corresponding to the variable combination from among the plurality of generated bin masks, and generate number data with the variable combination by using multiplication between the selected bin masks.

4. The electronic device of claim 1, wherein the bin mask comprises a plurality of slots, and

each of the plurality of slots comprises a plurality of sub slots comprising data on whether one variable value is present,

wherein the processor is configured to generate one bin mask on each of the homomorphic ciphertexts, and generate number data with the variable combination by using sub slots in the bin mask which correspond to the variable combination from among the generated bin masks.

5. The electronic device of claim 4, wherein the plurality of sub slots is configured to be disposed in one slot with a preset bit distance.

6. The electronic device of claim 1, wherein the processor is configured to join a first homomorphic ciphertext and a second homomorphic ciphertext into one homomorphic ciphertext, wherein the first homomorphic ciphertext and the second homomorphic ciphertext each comprises a plurality of data on a same feature.

7. The electronic device of claim 6, wherein the processor is configured to use a first position data in the first homomorphic ciphertext and a second position data in the

second homomorphic ciphertext on common data in the first homomorphic ciphertext and the second homomorphic ciphertext to join the first homomorphic ciphertext and the second homomorphic ciphertext into one homomorphic ciphertext.

8. The electronic device of claim 7, wherein the processor is configured to compare, based on the plurality of data comprised in the first and second homomorphic ciphertexts, wherein the plurality of data of each ciphertext is encrypted with a one-direction encryption scheme using a preset common key, and position data in a homomorphic ciphertext on the encrypted data being input, encrypted data on the first homomorphic ciphertext with encrypted data on the second homomorphic ciphertext, and check the first position data and the second position data which comprise common data between the two homomorphic ciphertexts.

9. A method of processing ciphertext on a homomorphic ciphertext, the method comprising:

storing a plurality of homomorphic ciphertexts, wherein each of the plurality of homomorphic ciphertexts stores a plurality of variable data in an encrypted state;

receiving an instruction to perform an operation on the plurality of homomorphic ciphertexts;

generating a bin mask corresponding to a variable combination and having different variable data classified for each of the plurality of homomorphic ciphertexts;

generating an encrypted number data corresponding to the variable combination by using the generated bin mask and the stored plurality of variable data; and

outputting the encrypted number data.

10. The method of claim 9, wherein each of the plurality of homomorphic ciphertexts comprises a plurality of slots, and

each of the plurality of slots comprises one variable data.

11. The method of claim 9, wherein the bin mask comprises a plurality of slots, and

each of the plurality of slots comprises data on whether one variable value is present,

wherein generating the bin mask comprises generating a plurality of bin masks for each variable data comprised in the homomorphic ciphertext with respect to each homomorphic ciphertext,

wherein generating the encrypted number data comprises selecting bin masks corresponding to the variable combination from among the plurality of generated bin masks, and using multiplication between the selected bin masks to generate number data with the variable combination.

12. The method of claim 9, wherein the bin mask comprises a plurality of slots, and

each of the plurality of slots comprises a plurality of sub slots comprising data on whether one variable value is present,

wherein generating the bin mask comprises generating one bin mask with respect to each of the homomorphic ciphertexts, and

wherein generating the encrypted number data comprises using sub slots in the bin mask corresponding to the variable combination from among the generated bin masks to

generate number data with the variable combination.

13. The method of claim 12, wherein the plurality of sub slots is configured to be disposed in one slot with a preset bit distance.

14. The method of claim 9, further comprising:

joining a first homomorphic ciphertext and a second homomorphic ciphertext into one homomorphic ciphertext, wherein the first homomorphic ciphertext and the second homomorphic ciphertext each comprises a plurality of data on a same feature.

15. The method of claim 14, wherein the joining comprises using a first position data in the first homomorphic ciphertext and a second position data in the second homomorphic ciphertext on common data in the first homomorphic ciphertext and the second homomorphic ciphertext, and joining the first homomorphic ciphertext and the second homomorphic ciphertext into one homomorphic ciphertext.

16. The method of claim 15, wherein the joining comprises comparing, based on the plurality of data comprised in the first and second homomorphic ciphertexts, wherein the plurality of data of each ciphertext is encrypted with a one-direction encryption scheme using a preset common key, and position data in a homomorphic ciphertext on the encrypted data being input, encrypted data on the first homomorphic ciphertext with encrypted data on the second homomorphic ciphertext, and checking the first position data and the second position data which comprise common data between the two homomorphic

ciphertexts.

17. A computer readable recording medium comprising a program for executing a ciphertext processing method, the method comprising:

storing a plurality of homomorphic ciphertexts, wherein each of the plurality of homomorphic ciphertexts stores a plurality of variable data in an encrypted state;

receiving an instruction to perform an operation on the plurality of homomorphic ciphertexts;

generating a bin mask corresponding to a variable combination and having different variable data classified for each of the plurality of homomorphic ciphertexts;

generating an encrypted number data corresponding to the variable combination by using the bin mask; and

outputting the encrypted number data.