What is claimed is:

1. A data comparison device (300) comprising:

a processor (501) and a memory (502), wherein

the memory holds first encrypted data in which a first plaintext is encrypted and second encrypted data in which a second plaintext is encrypted,

for the first plaintext divided into a plurality of blocks, the first encrypted data is data generated by executing processing including encryption of each of the plurality of blocks and shuffling of the plurality of blocks, wherein the first encrypted data encryption and shuffling of the blocks includes, where the first plaintext includes *n* blocks:

(i) concatenating an element, m_n, of the first plaintext with a first secret key (sk_1) to generate a first concatenation, and obtaining a first hash value of the first concatenation (S1303a),

(ii) concatenating the first hash value with a random number (r) to generate a second concatenation, and obtaining a second hash value of the second concatenation (S1304a),

(iii) performing an exclusive or operation of the second hash value and the element, m_n, of the first plaintext, and storing the calculation result,

(iv) shuffling the position of the calculation result using a second secret key (sk_2),

(v) repeating steps (i) to (iv) for each element of the respective plain text from a most to least significant block and

(vi) storing the shuffled calculation results together with the random number (r) as the respective encrypted data;

for the second plaintext divided into the plurality of blocks, the second encrypted data is data generated by executing processing including encryption of each of the plurality of blocks, wherein the second encrypted data is an encrypted query generated by a processing which includes, for a query comprising *n* bits:

(i) inverting an element (w_i) of the query to generate a different element (w_i');

(ii) concatenating the different element with the first secret key (sk_1), and obtaining a third hash value from the concatenation;

(iii) shuffling the position of the third hash value using the second secret key (sk_2), and storing the third hash at the shuffled position; and

(iv) repeating steps (i) to (iii) for each element of the query from a most to least significant bit;

in at least one of the first encrypted data and the second encrypted data, a value of the at least one plaintext is embedded as a value indicating a magnitude comparison result,

and

the processor compares blocks at the same position before shuffling of the first encrypted data and the second encrypted data based on the embedded value of the at least one plaintext and determines a magnitude relationship between the first plaintext and the second plaintext

wherein comparing the blocks includes:

(i) extracting the random number (r) from the first encrypted data;

(ii) extracting an element, c_i, from the first encrypted data and an element, q_i, from the second encrypted data;

(iii) concatenating q_i with the random number, and generating a hash H_r from the concatenation;

(iv) performing an exclusive or operation of the hash H_r and the element c_i;

(v) determining the result of the exclusive or operation, and if it is determined that the result is 1, output that the first plaintext is larger than the second plaintext, and if it is determined that the result is 0, output that the first plaintext is smaller than the second plaintext, otherwise, output that the first plaintext and second plaintext are equal.

2. The data comparison device according to claim 1, wherein

in the encryption of each of the plurality of blocks in at least one of the first encrypted data and the second encrypted data, a value of each block of the one plaintext is converted to a different value.

3. The data comparison device according to claim 1, wherein

the processor

obtains the random number from the first encrypted data,

calculates an exclusive OR of a value obtained by inputting the bit and the random number to the second function and a bit at the same position before shuffling of the first encrypted data, for the bits of the second encrypted data, and

determines a magnitude relationship between the first plaintext and the second plaintext based on the calculated exclusive OR.

4. The data comparison device according to claim 2, wherein

the first plaintext and the second plaintext are base-p numeric strings,

p is an integer of 3 or more,

for the first plaintext divided every predetermined number of digits, the first encrypted data is data generated by executing processing including encryption in which for each digit value, a value of the digit, a value of a digit higher than the digit, and a value of the secret key input to a first function are obtained, and the obtained values and a first random number are input to a second function, and shuffling of the first plaintext in units of the predetermined number of digits,

the first encrypted data includes the first random number, and

for the second plaintext divided every predetermined number of digits, the second encrypted data is data generated by executing processing including encryption in which for each digit value, a value of the digit, a value of a digit higher than the digit, and the secret key are input to the first function or a newly generated random number is set as the value of the digit.

5. The data comparison device according to claim 4, wherein

the processor

obtains the first random number from the first encrypted data,

calculates an exclusive OR of a value obtained by

inputting the bit and the random number to the second function and the same digit before shuffling of the first encrypted data, for the bits of the second encrypted data, and

determines a magnitude relationship between the first plaintext and the second plaintext based on the calculated exclusive OR.

6. A data comparison system comprising:

a data comparison device (300);

a first encrypted data generation device (100) that encrypts a first plaintext (111); and

a second encrypted data generation device (200) that encrypts a second plaintext, wherein

the first encrypted data generation device

divides the first plaintext into a plurality of blocks, and

generates a first encrypted data by executing processing including encryption of the first plaintext for each of the plurality of blocks and shuffling of the plurality of blocks, wherein the first encrypted data encryption and shuffling of the blocks includes, where the first plaintext includes *n* blocks:

(i) concatenating an element, m_n, of the first plaintext with a first secret key (sk_1) to generate a first concatenation, and obtaining a first hash value of the first

concatenation (S1303a),

(ii) concatenating the first hash value with a random number (r) to generate a second concatenation, and obtaining a second hash value of the second concatenation (S1304a),

(iii) performing an exclusive or operation of the second hash value and the element, m_n, of the first plaintext, and storing the calculation result,

(iv) shuffling the position of the calculation result using a second secret key (sk_2),

(v) repeating steps (i) to (iv) for each element of the respective plain text from a most to least significant block and

(vi) storing the shuffled calculation results together with the random number (r) as the respective encrypted data;

the second encrypted data generation device

divides the second plaintext into the plurality of blocks,

generates a second encrypted data by executing processing including encryption of the second plaintext for each of the plurality of blocks, wherein the second encrypted data is an encrypted query generated by a processing which includes, for a query comprising *n* bits:

(i) inverting an element (w_i) of the query to

generate a different element (w_i');

(ii) concatenating the different element with the first secret key (sk_1), and obtaining a third hash value from the concatenation;

(iii) shuffling the position of the third hash value using the second secret key (sk_2), and storing the third hash at the shuffled position; and

(iv) repeating steps (i) to (iii) for each element of the query from a most to least significant bit;

when at least one of processing of embedding a value of the first plaintext as a value indicating a magnitude comparison result in the generation of the first encrypted data by the first encrypted data generation device and processing of embedding a value of the second plaintext as a value indicating a magnitude comparison result in the generation of the second encrypted data by the second encrypted data generation device is executed,

the first encrypted data generation device transmits the first encrypted data to the data comparison device and

the second encrypted data generation device transmits the second encrypted data to the data comparison device, and

the data comparison device compares blocks at the same position before shuffling of the first encrypted data and the second encrypted data based on the embedded value of the at least one plaintext and determines a magnitude relationship

between the first plaintext and the second plaintext based on a value indicating the magnitude comparison result, wherein comparing the blocks includes:

(i) extracting the random number (r) from the first encrypted data;

(ii) extracting an element, c_i, from the first encrypted data and an element, q_i, from the second encrypted data;

(iii) concatenating q_i with the random number, and generating a hash H_r from the concatenation;

(iv) performing an exclusive or operation of the hash H_r and the element c_i;

(v) determining the result of the exclusive or operation, and if it is determined that the result is 1, output that the first plaintext is larger than the second plaintext, and if it is determined that the result is 0, output that the first plaintext is smaller than the second plaintext, otherwise, output that the first plaintext and second plaintext are equal.

7. A data comparison method executed by a data comparison device (300), wherein

the data comparison device includes a processor (501) and a memory (502),

the memory holds first encrypted data in which a first plaintext is encrypted and second encrypted data in which a

second plaintext is encrypted,

for the first plaintext divided into a plurality of blocks, the first encrypted data is data generated by executing processing including encryption of each of the plurality of blocks and shuffling of the plurality of blocks, wherein the first encrypted data encryption and shuffling of the blocks includes, where the first plaintext includes *n* blocks:

(i) concatenating an element, m_n, of the first plaintext with a first secret key (sk_1) to generate a first concatenation, and obtaining a first hash value of the first concatenation (S1303a),

(ii) concatenating the first hash value with a random number (r) to generate a second concatenation, and obtaining a second hash value of the second concatenation (S1304a),

(iii) performing an exclusive or operation of the second hash value and the element, m_n, of the first plaintext, and storing the calculation result,

(iv) shuffling the position of the calculation result using a second secret key (sk_2),

(v) repeating steps (i) to (iv) for each element of the respective plain text from a most to least significant block and

(vi) storing the shuffled calculation results together with the random number (r) as the respective encrypted

data;

for the second plaintext divided into the plurality of blocks, the second encrypted data is data generated by executing processing including encryption of each of the plurality of blocks, wherein the second encrypted data is an encrypted query generated by a processing which includes, for a query comprising *n* bits:

(i) inverting an element (w_i) of the query to generate a different element (w_i');

(ii) concatenating the different element with the first secret key (sk_1), and obtaining a third hash value from the concatenation;

(iii) shuffling the position of the third hash value using the second secret key (sk_2), and storing the third hash at the shuffled position; and

(iv) repeating steps (i) to (iii) for each element of the query from a most to least significant bit;

in at least one of the first encrypted data and the second encrypted data, a value of the at least one plaintext is embedded as a value indicating a magnitude comparison result, and

the processor compares blocks at the same position before shuffling of the first encrypted data and the second encrypted data based on the embedded value of the at least one plaintext and determines a magnitude relationship between the first

plaintext and the second plaintext, wherein comparing the blocks includes:

(i) extracting the random number (r) from the first encrypted data;

(ii) extracting an element, c_i, from the first encrypted data and an element, q_i, from the second encrypted data;

(iii) concatenating q_i with the random number, and generating a hash H_r from the concatenation;

(iv) performing an exclusive or operation of the hash H_r and the element c_i;

(v) determining the result of the exclusive or operation, and if it is determined that the result is 1, output that the first plaintext is larger than the second plaintext, and if it is determined that the result is 0, output that the first plaintext is smaller than the second plaintext, otherwise, output that the first plaintext and second plaintext are equal.