Secure Two-party Computation for Equality of Bilateral Data Based on Pinhole Camera Model

Changfeng Gui [a] Mengsi Han [b] Wei He [c] Zhantong Xie [a]

[a] Faculty of Science and Technology, University of Macau, Macau, P. R. China
[b] Hunan Zeta Technology Limited, Changsha, P. R. China
[c] School of Mathematics and Statistics, Central South University, Changsha, P. R. China
changfenggui@um.edu.mo

Abstract

Secure two-party computation is always the focus in the international cryptographic community, and establishing a secure two-party computation for determining the equality of data is a fundamental and crucial task within this field. There is still substantial room for improvement in terms of efficiency and security, so it is worth exploring further. In this paper, two theorems are proved from camera imaging of straight lines on a flat surface and the equality judgement of data between the two-party is innovatively converted into judging whether the inner products of the vectors owned by the two parties are zeroes. Based on the specific property of these vectors, a security protocol is established to determine whether those inner products are zeroes, which realizes the protocol scheme of judging the data equality of two-party under the semi-honest model and analyses its security theoretically. The scheme needs key stream for dynamic data encoding, since only basic arithmetic operations (integer addition, subtraction and multiplication) and a final modulus operation are required after encode without high-order and decimal operations, and the communication complexity is O(1), which greatly improves the efficiency. This scheme may not fit for malicious model, we could use the existing agreements of malicious model to determine whether those inner products are zeroes, by the conversion in this paper we also could complete the equality judgement.

Key words: Secure Multi-party Computation, Data equality test, Semi-honest model, key steam

1 Introduction

Secure two-party computation (2PC) is a field of cryptography involving two mutually distrusting parties who wish to securely compute an arbitrary function on their private input so that each of them can learn some private outputs. This field is proposed by Yao [1], who proposed a protocol for the millionaires' problem, that is, to determine which of the two participants is richer, so that no information about a party's amount

Z. Zheng and T. Wang (eds.), *Proceedings of the International Academic Summer Conference on Number Theory and Information Security (NTIS 2023)*, Advances in Physics Research 9,
https://doi.org/10.2991/978-94-6463-463-1_2

of assets is leaked to the other party, specifically for the semi-honest model where both parties are assumed to follow the prescribed protocol. It lays the groundwork for subsequent research of cryptography and privacy protection. In [2], Cleve demonstrated an impossibility by revealing that certain functions could not be computed with complete fairness unless there was an honest majority. Subsequently, Goldreich et al. [3] presented a solution within the malicious setting, where one of the parties might deviate arbitrarily from the prescribed protocol, offering a general theoretical resolution to issues in secure multi-party computation and its security. Nonetheless, the efficiency of this protocol was hindered as it relied on public-key operations for each Boolean gate in the circuit describing the function to compute. Subsequent extensive research in the field of 2PC has led to the development of practically efficient protocols that are secure against both semi-honest and malicious adversaries [4]- [9]. These studies not only propelled the theoretical development of 2PC but also provided practical tools for real-world applications like e-commerce and data mining.

The private comparison of the equal information is a very important problem. The development of 2PC protocols also includes various solutions based on comparison problems, such as Cachin's third-party based GT protocol and the GT protocol developed by Ioannidis and Grama using Oblivious Transfer (OT). Boudot [10] proposed a protocol addressing the socialist millionaires' problem, wherein two millionaires seek to determine if they are equally rich. For active adversaries, Lo [11] shows that the equality function cannot be securely evaluated between two (all-powerful) parties. However, if the additional assumptions are made, the goal can be obtained. Bogdanov et al. [12] present a provably secure and efficient general-purpose computation system to address sensitive data. Damgard [13] used the additional assumption of bounded quantum storage to achieve the secure evaluation of the equality function. It is possible that an additional party as considered allow for the secure computation of the equality as well. Yang et al. [14] proposed an efficient protocol for two-party quantum private comparison with the TP's help and the hash function. However, the hash function cannot guarantee a one-to-one mapping, that is, when two characters are not equal, the corresponding hash values may be equal, which makes the protocol unable to guarantee the correctness of the judgement theoretically. The literature [15], with the help of a third party (TP), proposes an effective equivalent information comparison protocol. Assuming the TP is semi-honest, that is, the TP faithfully executes the protocol, records all intermediate calculations, and may attempt to steal the player's private input from the record, but he cannot be corrupted by the opponent. In [16], the problem of personalized multi-keyword ranking search for encrypted data in cloud computing is studied and solved for the first time under the premise of privacy protection. Furthermore, Couteau [17] introduced new protocols for securely computing the greater than and the equality between two parties, which is very suitable for large-scale secure computing protocols, including security comparison (SC) and equality testing.

From a large number of research literatures, the transformation of the research on secure 2PC from traditional secure two-party protocols to quantum encryption shows that the problem still has further improvement needs and hopes to find solutions from new research methods. To determine whether the data of two-party are equal, the data of two-party can be converted into the space pinhole coordinates of the third three-dimensional integer type with non-zero component by injective method according to the convention. Then, whether the data of the two-party are equal is converted into whether the three-dimensional integral type space pinhole coordinates of the two-party are the same. This paper innovatively finds a secure two-party protocol that converts the judgement of whether the data of two-party are equal to the inner product of vectors from the principle of pinhole imaging of plane linear. We first analyze the principle of pinhole imaging of planar straight lines and important conclusions related to the protocol, and then apply the relevant conclusions to determine whether the two numbers are equal, and establish a secure two-party calculation to determine whether the two numbers are equal to the vector inner product. Because the data does not appear separately in each component of the vector, it can well protect the security of the data.

2 Pinhole imaging principle of straight lines in a plane

2.1 Principle of pinhole imaging

First establish a coordinate system $\{O; \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ in the object space, referred as the world coordinate system. Let $P_0(x_0, y_0, z_0)$be the world coordinate of the camera's pinhole, The camera axis line passing through the pinhole in the z' direction is $\boldsymbol{k}' = (l_3, m_3, n_3)$, which points from the pinhole towards the outside of the camera, perpendicular to the receiving screen. The plane perpendicular to the camera axis line at the pinhole is called the camera plane. Let the direction vector of x' be $\boldsymbol{i}' = (l_1, m_1, n_1)$, representing the horizontal direction of the camera plane, and consequently, the vertical direction of the camera plane is $\boldsymbol{j}' = \boldsymbol{k}' \times \boldsymbol{i}' = (l_2, m_2, n_2)$. Assume the distance from the pinhole to the plane where the receiving screen is located is d, and that the receiving screen's plane is perpendicular to the axis line.

Let $P(x, y, z)$ be any point in the world coordinate system, and (x', y', z') be the corresponding point in the pinhole coordinate system. Then, we have

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

Letting $\boldsymbol{x} = [x, y, z]^T$, $\boldsymbol{x}' = [x', y', z']^T$, $\boldsymbol{x}_0 = [x_0, y_0, z_0]^T$, $R = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix}$, then the above equation can be written as $\boldsymbol{x} = R\boldsymbol{x}' + \boldsymbol{x}_0$

Let $P'(u', v', -d)$ be the correspond point on the receiving screen after pinhole imaging of $P(x, y, z)$, then we have $\frac{u'}{x'} = \frac{v'}{y'} = \frac{-d}{z'}$, denote $\boldsymbol{u}' = [u', v', 1]^T$,

$$\Lambda = \begin{bmatrix} -d & 0 & 0 \\ 0 & -d & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

then $\boldsymbol{u}' = \frac{1}{z'}\Lambda\boldsymbol{x}' = \frac{1}{z'}\Lambda(R^T\boldsymbol{x} - R^T\boldsymbol{x}_0)$, simplifying it we can get $\boldsymbol{u}' = \frac{1}{z'}\Lambda\boldsymbol{x}' =$

$\frac{1}{z'}\Lambda \begin{bmatrix} R^T & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}$, here $\boldsymbol{t} = -R^T \boldsymbol{x}_0$.

Let the pixel coordinate $P'(u', v', -d)$ corresponding to the coordinate $P''(u, v)$ on the receiving screen as the following formula: $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha' & \gamma' & u_0 \\ s' & \beta' & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$,here u represents the row and v represents the column in the pixel coordinates.

Denote $K = \begin{bmatrix} \alpha' & \gamma' & u_0 \\ s' & \beta' & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ and $z_p = l_3 x + m_3 y + n_3 z + z_0'$, then by rearranging the above formula, we get $\boldsymbol{u} = \frac{1}{z_p} K \Lambda \begin{bmatrix} R^T & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}$, where $\boldsymbol{u} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$.

By orthogonal decomposition of

$K\Lambda = \begin{bmatrix} -d\alpha' & -d\gamma' & u_0 \\ -ds' & -d\beta' & v_0 \\ 0 & 0 & 1 \end{bmatrix}$,

there exist $K' = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ and θ which satisfy

$$\begin{bmatrix} -d\alpha' & -d\gamma' & u_0 \\ -ds' & -d\beta' & v_0 \\ 0 & 0 & 1 \end{bmatrix} = K' \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Denote $\begin{bmatrix} l_1' & m_1' & n_1' & x_0'' \\ l_2' & m_2' & n_2' & y_0'' \\ l_3' & m_3' & n_3' & z_0' \end{bmatrix}$ as

$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_1 & m_1 & n_1 & x_0' \\ l_2 & m_2 & n_2 & y_0' \\ l_3 & m_3 & n_3 & z_0' \end{bmatrix}$,

then it is not difficult to obtain that

$\begin{pmatrix} l_1' & m_1' & n_1' \end{pmatrix}^T, \begin{pmatrix} l_2' & m_2' & n_2' \end{pmatrix}^T$

are the world coordinates of the vectors obtained by rotating $\boldsymbol{i}', \boldsymbol{j}'$ counter-clockwise θ within the horizontal plane of the pinhole coordinate system.

For the ease of subsequent discussion, the two vectors after rotation are still denoted as $\boldsymbol{i}' = \begin{pmatrix} l_1 & m_1 & n_1 \end{pmatrix}^T, \boldsymbol{j}' = \begin{pmatrix} l_2 & m_2 & n_2 \end{pmatrix}^T$. In the new pinhole coordinate system $\{P_0; \boldsymbol{i}', \boldsymbol{j}', \boldsymbol{k}'\}$, let (x_0'', y_0'', z_0') be the coordinates of the world coordinate system's origin in this system. For convenience, we still denote $(x_0', y_0', z_0') = (x_0'', y_0'', z_0')$, $\begin{bmatrix} R^T & \boldsymbol{t} \end{bmatrix} = \begin{bmatrix} l_1' & m_1' & n_1' & x_0'' \\ l_2' & m_2' & n_2' & y_0'' \\ l_3' & m_3' & n_3' & z_0' \end{bmatrix}$ and $K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$.

Using the agreed notation, leading to the following vector form pinhole imaging formula:

$$\boldsymbol{u} = \frac{1}{z_p} K \begin{bmatrix} R^T & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \tag{2.1}$$

Assuming there is a receiving screen shifted outward by one unit on the horizontal plane of the new pinhole coordinate system, then the coordinate of the pinhole imaging point $\boldsymbol{u} = \begin{pmatrix} u & v & 1 \end{pmatrix}^T$ satisfies $\boldsymbol{u'} = \frac{1}{z_p}\begin{bmatrix} R^T & \boldsymbol{t} \end{bmatrix}\begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}$. After undergoing rotation, translation, and stretching transformations $K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, the pixel coordinate (u, v) satisfies the formula $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$, which collectively represents Imaging Formula (1). This understanding is equivalent to the principle of pinhole imaging.

2.2 Imaging lines that correspond to straight lines

Let $P_1(x_1, y_1, y_1)$ be a fixed point , and $\boldsymbol{v} = (l, m, n)^T$ be the direction vector of a line passing through P_1. For any point $P(x, y, z)$ on this line, $P(x, y, z)$ satisfies the equation: $\frac{x-x_1}{l} = \frac{y-y_1}{m} = \frac{z-z_1}{n}$.

Let $P''(u, v)$ be the pixel coordinates obtained using the pinhole imaging formula (1). Denote $a = (\boldsymbol{v}, \overrightarrow{P_0P_1}, \boldsymbol{i'})$, $b = (\boldsymbol{v}, \overrightarrow{P_0P_1}, \boldsymbol{j'})$, $c = (\boldsymbol{v}, \overrightarrow{P_0P_1}, \boldsymbol{k'})$, according to the equivalent principle of pinhole imaging, the corresponding imaging line equation for the above spatial line is:

$$(a, b, c)K^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

Since the three components of (a, b, c) are the inner products of $\boldsymbol{v} \times \overrightarrow{P_0P_1}$ with $\boldsymbol{i'}, \boldsymbol{j'}, \boldsymbol{k'}$ respectively, in the world coordinate system, $(\boldsymbol{v} \times \overrightarrow{P_0P_1})R$ precisely corresponds to vector (a, b, c). Thus, the line equation in the pixel coordinate system can be simplified to $(\boldsymbol{v} \times \overrightarrow{P_0P_1})R)K^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$.

Here $\boldsymbol{v} \times \overrightarrow{P_0P_1} = (\boldsymbol{v} \times \overrightarrow{P_0P_1})E$
$= ((\boldsymbol{v}, \overrightarrow{P_0P_1}, \boldsymbol{i}), (\boldsymbol{v}, \overrightarrow{P_0P_1}, \boldsymbol{j}), (\boldsymbol{v}, \overrightarrow{P_0P_1}, \boldsymbol{k}))$.

Assuming that the pinhole does not originate from the horizontal plane of the world coordinate system. Therefore, the third component of its world coordinates is not zero, and this assumption is followed throughout. Suppose there are vectors of three lines on a plane passing through the origin of the world coordinate system and their respective points denoted as $\boldsymbol{v}_i, P_i, i = 1, 2, 3$. Let $M_1 = \begin{bmatrix} \boldsymbol{v}_1 \times \overrightarrow{P_{01}P_1} \\ \boldsymbol{v}_2 \times \overrightarrow{P_{01}P_2} \\ \boldsymbol{v}_3 \times \overrightarrow{P_{01}P_3} \end{bmatrix}$ be an invertible matrix, where P_{01} is the pinhole coordinate.

Additionally, consider the vectors of three other lines on the previously discussed plane and their respective points as $\boldsymbol{v}_i', P_i', i = 1,2,3$. Let $M_2 = \begin{bmatrix} \boldsymbol{v}_1' \times \overrightarrow{P_{02}P_1'} \\ \boldsymbol{v}_2' \times \overrightarrow{P_{02}P_2'} \\ \boldsymbol{v}_3' \times \overrightarrow{P_{02}P_3'} \end{bmatrix}$ be an invertible matrix, where P_{02} is the pinhole coordinate.

Theorem 1. *Assume M_1, M_2 are invertible, the matrix $M_2M_1^{-1}$ is calculated as mentioned above.*

1) Let $\boldsymbol{v}_i, P_i, i = 1,2,3$ and $\boldsymbol{v}_i', P_i', i = 1,2,3$ be fixed,
and the pinhole coordinates P_{01}, P_{02} be collinear with the world coordinate origin. If perturbations are applied to the two pinhole coordinates while maintaining this collinearity and fixed ratio, then the resulting matrix remains unchanged;
Especially, if $P_{01} = P_{02}$, then $M_2M_1^{-1}$ is independent of P_{01} .

2) Let $\boldsymbol{v}_i, P_i, i = 1,2,3$ and $\boldsymbol{v}_i', P_i', i = 1,2,3$ be fixed, and P_{01}, P_{02} be the pinhole coordinates. If P_{01} is unchanged and P_{02} is perturbed, then the resulting matrix will necessarily change;

3) Let $\boldsymbol{v}_i, P_i, i = 1,2,3$ and $\boldsymbol{v}_i', P_i', i = 1,2$ be fixed,$\boldsymbol{v}_i', i = 1,2$ be non-collinear, and , P_{01}, P_{02} be pinhole coordinates. If P_{01} is unchanged and P_{02} is perturbed, then the resulting matrix will necessarily change. In this case, M_2 is a two-row, three-column matrix;

4) Let $\boldsymbol{v}_i, P_i, i = 1,2,3$ and $\boldsymbol{v}_i', P_i', i = 1,2,3$ be fixed, and $\boldsymbol{v}_i, i = 2,3$ be non-collinear, with and P_{01}, P_{02} as the pinhole coordinates. Then, if P_{01} remains constant and P_{02} is perturbed, then the resulting matrix $M_2M_1^{-1}(:, 1:2)$ will necessarily change;

Proof: (1) Since the concepts of co-planarity, collinearity, fixed ratio, and identical points do not change through orthogonal transformations , and since $(M_2R)(M_1R)^{-1} = M_2M_1^{-1}$, the discussion of $\boldsymbol{v}_i, P_i, i = 1,2,3$ and $\boldsymbol{v}_i', P_i', i = 1,2,3$ from a general plane passing through the origin can be transformed into a discussion from the horizontal plane in the world coordinate system.

As $M_1 = \begin{bmatrix} (\boldsymbol{v}_1, \overrightarrow{P_{01}P_1}, \boldsymbol{i}) & (\boldsymbol{v}_1, \overrightarrow{P_{01}P_1}, \boldsymbol{j}) & (\boldsymbol{v}_1, \overrightarrow{P_{01}P_1}, \boldsymbol{k}) \\ (\boldsymbol{v}_2, \overrightarrow{P_{01}P_2}, \boldsymbol{i}) & (\boldsymbol{v}_2, \overrightarrow{P_{01}P_2}, \boldsymbol{j}) & (\boldsymbol{v}_2, \overrightarrow{P_{01}P_2}, \boldsymbol{k}) \\ (\boldsymbol{v}_3, \overrightarrow{P_{01}P_3}, \boldsymbol{i}) & (\boldsymbol{v}_3, \overrightarrow{P_{01}P_3}, \boldsymbol{j}) & (\boldsymbol{v}_3, \overrightarrow{P_{01}P_3}, \boldsymbol{k}) \end{bmatrix}$,

let $\boldsymbol{v}_i, i = 1,2,3$ be parallel to the horizontal plane of the world coordinate system, with the world coordinates of $P_i, i = 1,2,3$ being $(x_i, y_i, 0)$. Let the coordinates of P_{01} be (x_{01}, y_{01}, z_{01}) , and let $\boldsymbol{v}_n = (\lambda_n, \mu_n, 0)$, then

$$(\boldsymbol{v}_n, \overrightarrow{P_{01}P_n}, \boldsymbol{i}) = \begin{vmatrix} \lambda_n & \mu_n & 0 \\ x_n - x_{01} & y_n - y_{01} & -z_{01} \\ 1 & 0 & 0 \end{vmatrix} = -z_{01}\mu_n,$$

$$(\boldsymbol{v}_n, \overrightarrow{P_{01}P_n}, \boldsymbol{j}) = \begin{vmatrix} \lambda_n & \mu_n & 0 \\ x_n - x_{01} & y_n - y_{01} & -z_{01} \\ 0 & 1 & 0 \end{vmatrix} = z_{01}\lambda_n,$$

$$(\boldsymbol{v}_n, \overrightarrow{P_{01}P_n}, \boldsymbol{k}) = \begin{vmatrix} \lambda_n & \mu_n & 0 \\ x_n - x_{01} & y_n - y_{01} & -z_{01} \\ 0 & 0 & 1 \end{vmatrix} = \lambda_n(y_n - y_{01}) - \mu_n(x_n - x_{01}).$$

For similar notations regarding $\boldsymbol{v}'_i, P'_i, i = 1, 2, 3$ and P_{02} , we have

$$M_2 = \begin{vmatrix} -z_{02}\mu'_1 & z_{02}\lambda'_1 & \lambda'_1(y'_1 - y_{02}) - \mu'_1(x'_1 - x_{02}) \\ -z_{02}\mu'_2 & z_{02}\lambda'_2 & \lambda'_2(y'_2 - y_{02}) - \mu'_2(x'_2 - x_{02}) \\ -z_{02}\mu'_3 & z_{02}\lambda'_3 & \lambda'_3(y'_3 - y_{02}) - \mu'_3(x'_3 - x_{02}) \end{vmatrix}. \text{ Since } M_1 = \begin{bmatrix} -z_{01}\mu_1 & z_{02}\lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ -z_{01}\mu_2 & z_{01}\lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ -z_{01}\mu_3 & z_{01}\lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{bmatrix}$$

$$\times \begin{bmatrix} 1 & 0 & -\frac{x_{01}}{z_{01}} \\ 0 & 1 & -\frac{y_{01}}{z_{01}} \\ 0 & 0 & 1 \end{bmatrix},$$

so $|M_1| = \begin{vmatrix} -z_{01}\mu_1 & z_{01}\lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ -z_{01}\mu_2 & z_{01}\lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ -z_{01}\mu_3 & z_{01}\lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}$

$= -z_{01}^2 \begin{vmatrix} \mu_1 & \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}$

After simplification, we get

$M_2(1, :)M_1^{-1}(:, 1) =$

$$\frac{\begin{bmatrix} -z_{02}\mu'_1 \\ z_{02}\lambda'_1 \\ (-(\frac{x_{01}}{z_{01}} - \frac{x_{02}}{z_{02}})z_{02}\mu'_1 + (\frac{y_{01}}{z_{01}} - \frac{y_{02}}{z_{02}})z_{02}\lambda'_1 \\ +\lambda'_1 y'_1 - \mu'_1 x'_1) \end{bmatrix}^T}{-z_{01}\begin{vmatrix} \mu_1 & \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}}$$

$$\times \begin{bmatrix} \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -z_{01}\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix} \end{bmatrix}$$

If P_{01} and P_{02} are collinear with the world coordinate origin, then $\exists \nu$ such that $(x_{01}, y_{01}, z_{01}) = \nu(x_{02}, y_{02}, z_{02})$. Since the pinhole coordinates are not on the horizontal plane here $z_{01} \neq 0$, thus $\nu \neq 0$.

Therefore, $-\frac{x_{01}}{z_{01}} + \frac{x_{02}}{z_{02}} = -\frac{\nu x_{02}}{\nu z_{02}} + \frac{x_{02}}{z_{02}} = 0$,$\frac{y_{01}}{z_{01}} - \frac{y_{02}}{z_{02}} = \frac{\nu y_{02}}{\nu z_{02}} - \frac{y_{02}}{z_{02}} = 0$

Consequently,

$M_2(1, :)M_1^{-1}(:, 1) =$

$$\frac{\begin{bmatrix} -\frac{z_{02}}{z_{01}}\mu'_1 \\ \frac{z_{02}}{z_{01}}\lambda'_1 \\ \lambda'_1 y'_1 - \mu'_1 x'_1 \end{bmatrix}^T}{\begin{vmatrix} \mu_1 & \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}} \begin{bmatrix} \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix} \end{bmatrix}$$

Thus $M_2(1,:)M_1^{-1}(:,1)$ remains unchanged under condition (1) of Theorem 1. Similarly, $M_2(i,:)M_1^{-1}(:,j), i,j=1,2,3$ also remains unchanged under condition (1) of Theorem 1.

(2) Given $M_2 = \begin{bmatrix} -z_{02}\mu'_1 & z_{02}\lambda'_1 & y'_1\lambda'_1 - x'_1\mu'_1 \\ -z_{02}\mu'_2 & z_{02}\lambda'_2 & y'_2\lambda'_2 - x'_2\mu'_2 \\ -z_{02}\mu'_3 & z_{02}\lambda'_3 & y'_3\lambda'_3 - x'_3\mu'_3 \end{bmatrix}$
$\times \begin{bmatrix} 1 & 0 & -\frac{x_{02}}{z_{02}} \\ 0 & 1 & -\frac{y_{02}}{z_{02}} \\ 0 & 0 & 1 \end{bmatrix}$,
when other conditions remain constant and only P_{02} is perturbed, then at least one of $\frac{x_{02}}{z_{02}}$,$\frac{y_{02}}{z_{02}}$, and z_{02} will change. Due to the invertible of M_2 , if z_{02} changes, then the first two columns of M_2 must have a changing component. Therefore, if z_{02} remains constant, then one of $\frac{x_{02}}{z_{02}}$ or $\frac{y_{02}}{z_{02}}$ must change. Consequently, changes, leading to the change of $\begin{bmatrix} 1 & 0 & -\frac{x_{02}}{z_{02}} \\ 0 & 1 & -\frac{y_{02}}{z_{02}} \\ 0 & 0 & 1 \end{bmatrix}$, and thus M_2 changes.

(3) Given

$$M_2 = \begin{bmatrix} -z_{02}\mu'_1 & z_{02}\lambda'_1 & \begin{pmatrix} y'_1\lambda'_1 - x'_1\mu'_1 \\ +(x_{02}\mu'_1 - y_{02}\lambda'_1) \end{pmatrix} \\ -z_{02}\mu'_2 & z_{02}\lambda'_2 & \begin{pmatrix} y'_2\lambda'_2 - x'_2\mu'_2 \\ +(x_{02}\mu'_2 - y_{02}\lambda'_1) \end{pmatrix} \end{bmatrix},$$

when other conditions remain constant and only P_{02} is perturbed, then at least one of x_{02} , y_{02} , and z_{02} will change. If z_{02} changes, the components of the first two columns of M_2 must change. Therefore, if either x_{02} or y_{02} changes, given the non-collinearity of $\boldsymbol{v}_i, i=1,2$,
$\begin{bmatrix} x_{02}\mu'_1 - y_{02}\lambda'_1 \\ x_{02}\mu'_2 - y_{02}\lambda'_2 \end{bmatrix} = \begin{bmatrix} \mu'_1 & -\lambda'_1 \\ \mu'_2 & -\lambda'_2 \end{bmatrix} \begin{bmatrix} x_{02} \\ y_{02} \end{bmatrix}$ must change.

Consequently, M_2 changes. With everything else remaining constant, M_1^{-1} remains unchanged, hence $M_2 M_1^{-1}$ changes.

(4) Let
$\alpha_1 = \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}, \alpha_2 = -\begin{vmatrix} \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}$
$\alpha_3 = \begin{vmatrix} \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \lambda_2 & y_2\lambda_2 - x_2\mu_2 \end{vmatrix}, \beta_1 = \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix},$
$\beta_2 = -\begin{vmatrix} \mu_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}, \beta_3 = \begin{vmatrix} \mu_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & y_2\lambda_2 - x_2\mu_2 \end{vmatrix},$
$\gamma_1 = -\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix}, \gamma_2 = \begin{vmatrix} \mu_1 & \lambda_1 \\ \mu_3 & \lambda_3 \end{vmatrix}, \gamma_3 = -\begin{vmatrix} \mu_1 & \lambda_1 \\ \mu_2 & \lambda_2 \end{vmatrix}.$

Let $m = \begin{vmatrix} \mu_1 & \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}$, it is not difficult to obtain

$$M_1^{-1} = \frac{1}{-z_{01}m} \begin{bmatrix} 1 & 0 & x_{01} \\ 0 & 1 & y_{01} \\ 0 & 0 & z_{01} \end{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \gamma_1 & \gamma_2 & \gamma_3 \end{bmatrix} \tag{2.2}$$

Therefore, $M_1^{-1}(:, 1:2)$
$= \frac{1}{-z_{01}m} \begin{bmatrix} \alpha_1 + x_{01}\gamma_1 & \alpha_2 + x_{01}\gamma_2 \\ \beta_1 + y_{01}\gamma_1 & \beta_2 + y_{01}\gamma_2 \\ z_{01}\gamma_1 & z_{01}\gamma_2 \end{bmatrix}$.
Let $x'' = \frac{x_{01}}{z_{01}}, y'' = \frac{y_{01}}{z_{01}}, z'' = \frac{1}{z_{01}}$, then
$M_1^{-1}(:, 1:2) = \frac{1}{-m} \begin{bmatrix} z''\alpha_1 + x''\gamma_1 & z''\alpha_2 + x''\gamma_2 \\ z''\beta_1 + y''\gamma_1 & z''\beta_2 + y''\gamma_2 \\ \gamma_1 & \gamma_2 \end{bmatrix}$.
Because M_1 is invertible and due to equation (2),
$\begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \\ \gamma_1 & \gamma_2 \end{bmatrix}$ has a rank of 2. $\boldsymbol{v}_i, i = 2, 3$ are non-collinear and $\gamma_1 \neq 0$, it follows by contradiction that either $\begin{vmatrix} \alpha_1 & \alpha_2 \\ \gamma_1 & \gamma_2 \end{vmatrix} \neq 0$ or $\begin{vmatrix} \beta_1 & \beta_2 \\ \gamma_1 & \gamma_2 \end{vmatrix} \neq 0$ must be true.
When P_{01} changes, (x'', y'', z'') must change. Let the change vector be $(\Delta x, \Delta y, \Delta z)$, then $(\Delta x, \Delta y, \Delta z)$ is not a zero vector. Assuming $M_1^{-1}(:, 1:2)$ remains unchanged when only P_{01} changes under other constant conditions, then $\begin{bmatrix} \Delta z\alpha_1 + \Delta x\gamma_1 & \Delta z\alpha_2 + \Delta x\gamma_2 \\ \Delta z\beta_1 + \Delta y\gamma_1 & \Delta z\beta_2 + \Delta y\gamma_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$,
i.e., $\begin{bmatrix} \gamma_1 & 0 & \alpha_1 \\ \gamma_2 & 0 & \alpha_2 \\ 0 & \gamma_1 & \beta_1 \\ 0 & \gamma_2 & \beta_2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$.

However, since $\begin{vmatrix} \alpha_1 & \alpha_2 \\ \gamma_1 & \gamma_2 \end{vmatrix} \neq 0$ or $\begin{vmatrix} \beta_1 & \beta_2 \\ \gamma_1 & \gamma_2 \end{vmatrix} \neq 0$, it follows that $\begin{bmatrix} \gamma_1 & 0 & \alpha_1 \\ \gamma_2 & 0 & \alpha_2 \\ 0 & \gamma_1 & \beta_1 \end{bmatrix} \neq 0$ or $\begin{bmatrix} \gamma_1 & 0 & \alpha_1 \\ \gamma_2 & 0 & \alpha_2 \\ 0 & \gamma_2 & \beta_2 \end{bmatrix} \neq 0$, hence $\begin{bmatrix} \gamma_1 & 0 & \alpha_1 \\ \gamma_2 & 0 & \alpha_2 \\ 0 & \gamma_1 & \beta_1 \\ 0 & \gamma_2 & \beta_2 \end{bmatrix}$ has a rank of 3. This contradicts the existence of a non-zero vector solution $(\Delta x, \Delta y, \Delta z)$ for $\begin{bmatrix} \gamma_1 & 0 & \alpha_1 \\ \gamma_2 & 0 & \alpha_2 \\ 0 & \gamma_1 & \beta_1 \\ 0 & \gamma_2 & \beta_2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} =$

$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$. Therefore, when P_{01} changes, $M_1^{-1}(:, 1:2)$ must change, and thus $M_2M_1^{-1}(:, 1:2)$ must change.

Theorem 2. *Assuming M_1, M_2 are invertible, the matrix $M_2M_1^{-1}$ is computed according to Theorem 1. Let $\boldsymbol{v}_i, P_i, i = 1, 2, 3$ and $\boldsymbol{v}'_i, P'_i, i = 1, 2, 3$ satisfy the previous requirements, and P_0 be the pinhole coordinates. Let $P_{01} = P_{02} = P_0$ calculate the matrix $\mathbf{Mat}_1 = M_2M_1^{-1}$ according to Theorem 1, and then calculate the matrix $\mathbf{Mat}_2$ with the actual P_{01} and P_{02} that need to be assessed for equality as pinhole coordinates. Then, $\mathbf{Mat}_1 = \mathbf{Mat}_2$ if and only if $P_{01} = P_{02}$.*

Proof: According to Theorem 1, when $P_{01} = P_{02}$, applying the same perturbations to the pinhole coordinates will still result in equality, satisfying the condition of Theorem 1(1). Thus, the calculated matrix remains unchanged under the same perturbations. Therefore, to determine whether P_{01} and P_{02} are equal, one can first agree on a set of pinhole coordinates P_0 and calculate a matrix $\mathbf{Mat}_1 = M_2M_1^{-1}$ according to Theorem 1. As this matrix remains unchanged under the same perturbations, it can be understood as being calculated with the same pinhole coordinates P_{01} according to Theorem 1, then by P_{01} and P_{02} according to Theorem 1 calculate the matrix $\mathbf{Mat}_2$. Thus, when $P_{01} = P_{02}$, it follows that $\mathbf{Mat}_1 = \mathbf{Mat}_2$, and by Theorem 1(2), when $P_{01} \neq P_{02}$,we have $\mathbf{Mat}_1 \neq \mathbf{Mat}_2$.

From Theorem 1(3), it is known that when $\boldsymbol{v}'_i, i = 1, 2$ are non-collinear, one can discuss the situation where M_2 is a two-row, three-column matrix. In this case, removing the invertibility condition of M_2, Theorem 2 still holds.

Theorem 1(4) shows that when $\boldsymbol{v}_i, i = 2, 3$ is not collinear, only the matrix $M_2M_1^{-1}(:, 1:2)$ is discussed, and theorem 2 still holds.

2.3 Preliminary approach to applying Theorem 2 in secure two-party computation

Suppose Alice has $P_0, P_{01}, \boldsymbol{v}_i, P_i, i = 1, 2, 3$, and Bob has P_0, P_{02} , $\boldsymbol{v}'_i$,$P'_i, i = 1, 2, 3$. Both parties first use an agreed set of identical pinhole coordinates P_0 as P_{01} and P_{02} to calculate a matrix value $\mathbf{Mat}_1 = M_2M_1^{-1}$ according to Theorem 1 . Then, both parties use their respective pinhole coordinates P_{01} and P_{02} to calculate a new matrix value $\mathbf{Mat}_2$ according to Theorem 1. According to Theorem 2, $\mathrm{Mat}_1 = \mathrm{Mat}_2$ if and only if $P_{01} = P_{02}$.

3 Protocol for Determining Equality of Data Between Two Parties Based on Vector Inner Product in Secure Two-party Computation Under Semi-Honest Model

Suppose Alice has three-dimensional integer data P_{01}, and Bob has three-dimensional integer data P_{02} . The objective is to design a secure bilateral protocol without third-party involvement to determine whether P_{01} and P_{02} are equal. Secure bilateral protocol under the semi-honest model:

Step 1. Alice or Bob sends the other party a randomly generated pinhole coordinate P_0 with integer components, hereP_0 serves as the public key for both parties;

Step 2. Alice and Bob independently generate random vector coordinates with integer components (the third component being zero) and spatial point coordinates$\boldsymbol{v}_i, P_i, i = 1, 2, 3$ and $\boldsymbol{v}'_i, P'_i, i = 1, 2, 3$ as private keys. $M_1^{-1}(:, 1) =$

$$\frac{1}{|M_1|}\begin{bmatrix} 1 & 0 & -\frac{x_{01}}{z_{01}} \\ 0 & 1 & -\frac{y_{01}}{z_{01}} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \begin{vmatrix} z_{01}\lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ z_{01}\lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -\begin{vmatrix} -z_{01}\mu_2 & y_2\lambda_2 - x_2\mu_2 \\ -z_{01}\mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} -z_{01}\mu_2 & z_{01}\lambda_2 \\ -z_{01}\mu_3 & z_{01}\lambda_3 \end{vmatrix} \end{bmatrix}$$

$$= \frac{1}{-z_{01}\begin{vmatrix} \mu_1 & \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}} \begin{bmatrix} z_{01} & 0 & x_{01} \\ 0 & z_{01} & y_{01} \\ 0 & 0 & z_{01} \end{bmatrix}$$

$$\times \begin{bmatrix} \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -z_{01}\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix} \end{bmatrix}$$

The calculation results here are fractions, with numerator and denominator computed separately. Alice calculates two matrices under her selected private key for pinhole coordinates P_0 and P_{01} , denoted as M_{11}, M_{12} . Bob calculates two matrices under his selected private key for pinhole coordinates P_0 and P_{02} , denoted as M_{21}, M_{22}

Step 3. Alice and Bob determine whether the calculated matrix $M_{21}M_{11}^{-1}$ is equal to $M_{22}M_{12}^{-1}$. According to Theorem 2, if they are not equal, then $P_{01} \neq P_{02}$; if they are equal, then $P_{01} = P_{02}$. Here, matrix operations follow the secure two-party computation of vector inner products.

Explanation: To determine whether $M_{21}M_{11}^{-1}$ is equal to $M_{22}M_{12}^{-1}$, consider that each component of M_{11}^{-1} has the same denominator $q_1 = -z_0 \begin{vmatrix} \mu_1 & \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \neq$

0, and each component of M_{12}^{-1} has the same denominator $q_2 = -z_{01} \begin{vmatrix} \mu_1 & \lambda_1 & y_1\lambda_1 - x_1\mu_1 \\ \mu_2 & \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \neq$ 0. Therefore, it suffices to determine whether $M_{21}\left(z_{01}M_{11}^{*}\right)$is equal to $M_{22}\left(z_0 M_{12}^{*}\right)$. Here, $M_{11}^{*}$and M_{12}^{*} are the adjoint matrices of the corresponding matrices, with each component corresponding to the inner product of two vectors. The equality of two inner product operations can be assessed by rearranging and combining the vector components to determine if the inner product is zero. Secure Two-party Computation of Vector Inner Products: Alice has a vector $(x_1, x_2, \cdots, x_n)$, and Bob has a vector $(y_1, y_2, \cdots, y_n)$.

Consider a secure bilateral protocol between them without the involvement of a third party.

Step 1: Alice randomly generates two integers $a \neq 0, r$ as a key and sends $(ax_1 + r, ax_2 + r, \cdots, ax_n + r)$ to Bob,

Step 2: Bob randomly generates a non-zero disturbance term b and calculates $c_1 = ax_1y_1b + ry_1b + ax_2y_2b + ry_2b + \cdots + ax_ny_nb + ry_nb$ and $c_2 = y_1b + y_2b + \cdots + y_nb$, which he sends to Alice.

Step 3: Alice calculates

$$c_1 - rc_2 = ax_1y_1b + ry_1b + ax_2y_2b + ry_2b + \cdots$$
$$+ax_ny_nb + ry_nb - r\left(y_1b + y_2b + \cdots + y_nb\right) =$$
$$a\left(x_1y_1 + x_2y_2 + \cdots + x_ny_n\right)b$$

Alice determines whether the inner product is zero based on whether the final result a$(x_1y_1 + x_2y_2 + \cdots + x_ny_n)$ b equals zero. For instance, when comparing the first row and first column of the matrix, Alice takes out

$$a_1 = z_0 M_{12}^{*}(:, 1)$$

$$= z_0 \begin{bmatrix} 1 & 0 & x_{01} \\ 0 & 1 & y_{01} \\ 0 & 0 & z_{01} \end{bmatrix} \begin{bmatrix} \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix} \end{bmatrix}$$

and

$$a_2 = z_{01} M_{11}^{*}(:, 1)$$

$$= z_{01} \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & z_0 \end{bmatrix} \begin{bmatrix} \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix} \end{bmatrix}.$$

After applying the two numbers $a \neq 0, r$, then obtains

$$az_0\begin{bmatrix} 1 & 0 & x_{01} \\ 0 & 1 & y_{01} \\ 0 & 0 & z_{01} \end{bmatrix}\begin{bmatrix} \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix} \end{bmatrix} + b$$

and

$$az_{01}\begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & z_0 \end{bmatrix}\begin{bmatrix} \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix} \\ -\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix} \end{bmatrix} + b,$$

then sends them to Bob.

Bob receives these six numbers and, by eliminating a and r , can derive four numbers. For ease of discussion, let

$$\alpha_1 = \begin{vmatrix} \lambda_2 & y_2\lambda_2 - x_2\mu_2 \\ \lambda_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix}, \beta_1 = \begin{vmatrix} \mu_2 & y_2\lambda_2 - x_2\mu_2 \\ \mu_3 & y_3\lambda_3 - x_3\mu_3 \end{vmatrix},$$

$$\gamma_1 = -\begin{vmatrix} \mu_2 & \lambda_2 \\ \mu_3 & \lambda_3 \end{vmatrix}.$$

Then, these six numbers are

$$\begin{bmatrix} az_0\alpha_1 + az_0x_{01}\gamma_1 + b \\ az_0\beta_1 + az_0y_{01}\gamma_1 + b \\ az_0z_{01}\gamma_1 + b \end{bmatrix}, \begin{bmatrix} az_{01}\alpha_1 + az_{01}x_0\gamma_1 + b \\ az_{01}\beta_1 + az_{01}y_0\gamma_1 + b \\ az_{01}z_0\gamma_1 + b \end{bmatrix}.$$

By eliminating a, r, Bob can obtain

$$b_1 = \frac{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_0\beta_1 - z_0y_{01}\gamma_1}{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_0z_{01}\gamma_1},$$
$$b_2 = \frac{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_{01}\alpha_1 - z_{01}x_0\gamma_1}{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_0z_{01}\gamma_1},$$
$$b_3 = \frac{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_{01}\beta_1 - z_{01}y_0\gamma_1}{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_0z_{01}\gamma_1},$$
$$b_4 = \frac{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_{01}z_0\gamma_1}{z_0\alpha_1 + z_0x_{01}\gamma_1 - z_0z_{01}\gamma_1}.$$

Thus, Bob can obtain four equations about Alice's information:

$$\begin{cases} b_1z_0\alpha_1 + b_1z_0x_{01}\gamma_1 - b_1z_0z_{01}\gamma_1 - z_0\alpha_1 - \\ z_0x_{01}\gamma_1 + z_0\beta_1 + z_0y_{01}\gamma_1 = 0 \\ b_2z_0\alpha_1 + b_2z_0x_{01}\gamma_1 - b_2z_0z_{01}\gamma_1 - z_0\alpha_1 - \\ z_0x_{01}\gamma_1 + z_{01}\alpha_1 + z_{01}x_0\gamma_1 = 0 \\ b_3z_0\alpha_1 + b_3z_0x_{01}\gamma_1 - b_3z_0z_{01}\gamma_1 - z_0\alpha_1 - \\ z_0x_{01}\gamma_1 + z_{01}\beta_1 + z_{01}y_0\gamma_1 = 0 \\ b_4z_0\alpha_1 + b_4z_0x_{01}\gamma_1 - b_4z_0z_{01}\gamma_1 - z_0\alpha_1 - \\ z_0x_{01}\gamma_1 + z_{01}z_0\gamma_1 = 0 \end{cases}$$

which is simplified as follow:

$$\begin{cases} (b_1-1)\,z_0\alpha_1+(b_1-1)\,z_0x_{01}\gamma_1- \\ b_1z_0z_{01}\gamma_1+z_0\beta_1+z_0y_{01}\gamma_1=0 \\ (b_2-1)\,z_0\alpha_1+(b_2-1)\,z_0x_{01}\gamma_1- \\ b_2z_0z_{01}\gamma_1+z_{01}\alpha_1+z_{01}x_0\gamma_1=0 \\ (b_3-1)\,z_0\alpha_1+(b_3-1)\,z_0x_{01}\gamma_1- \\ b_3z_0z_{01}\gamma_1+z_{01}\beta_1+z_{01}y_0\gamma_1=0 \\ (b_4-1)\,z_0\alpha_1+(b_4-1)\,z_0x_{01}\gamma_1- \\ b_4z_0z_{01}\gamma_1+z_{01}z_0\gamma_1=0 \end{cases}$$

Here, for Bob, α_1,β_1,γ_1, x_{01}, y_{01}, z_{01} are unknown variables.

By eliminating β_1 , Bob gets

$$\begin{cases} (b_1-1)\,z_0z_{01}\alpha_1+(b_1-1)\,z_0x_{01}z_{01}\gamma_1-b_1z_0z_{01}^2\gamma_1 \\ +z_0y_{01}z_{01}\gamma_1-(b_3-1)\,z_0^2\alpha_1-(b_3-1)\,z_0^2x_{01}\gamma_1 \\ +b_3z_0^2z_{01}\gamma_1-z_0z_{01}y_0\gamma_1=0 \\ (b_2-1)\,z_0\alpha_1+(b_2-1)\,z_0x_{01}\gamma_1-b_2z_0z_{01}\gamma_1+ \\ z_{01}\alpha_1+z_{01}x_0\gamma_1=0 \\ (b_4-1)\,z_0\alpha_1+(b_4-1)\,z_0x_{01}\gamma_1-b_4z_0z_{01}\gamma_1 \\ +z_{01}z_0\gamma_1=0 \end{cases}$$

Thus, further eliminating α_1, γ_1 will yield a nonlinear equation about x_{01}, y_{01}, z_{01}.

Recalling Theorem 1(4) and the corresponding Theorem 2, when $\boldsymbol{v}_i, i=2,3$ are non-collinear, calculations $M_2M_1^{-1}(:,1:2)$ with the agreed-upon P_0 and P_{01},P_{02} remain unchanged if and only if $P_{01}=P_{02}$.

Therefore, Alice only needs to send two columns of data, which means Bob can only obtain two equations about x_{01}, y_{01},z_{01}.

Note: Each time an inner product judgement is made, the receiver of the first encrypted vector can obtain an equation of the other party's data without parameters. If the other party actively attacks, three attempts can establish three equations, thereby using the obtained information to decrypt the other party's data.

To further enhance security and enable theoretically rigorous judgements, we can agree on a sufficiently large integer M .

Secure Two-party Computation of Vector Inner Product: Alice has a vector with integer components $(a_1,a_2,\cdots,a_n)$, and Bob has a vector with integer components $(b_1,b_2,\cdots,b_n)$. A secure protocol is established to enable Alice to determine whether the inner product is zero, without the involvement of a third party.

In the following discussion, Alice will generate random integers a, and Bob will generate random integers b. Here, the integer M needs to be greater than the theoretical maximum value of $a\,(a_1b_1+a_2b_2+\cdots+a_nb_n)\,b$ in the case of the agreed range of $a, b, (a_1,a_2,\cdots,a_n)$ and $(b_1,b_2,\cdots,b_n)$.

Step 1: Alice randomly generates an integer weight $a\neq 0$ and disturbance term r as well as an integer vector $(m_1,m_2,\cdots,m_n)$as a key, and randomly generates a sufficiently 1 large integer M that meets the previous requirements, and sends $(a\times a_1+r+m_1M, a\times a_2+r+m_2M,\cdots,$

$a\times a_n+r+m_nM)$ to Bob,

Step 2: Bob randomly generates a integer interference item $b\neq 0$ and calculates $c_1=aba_1b_1+rb_1b+m_1Mb_1b+aba_2b_2+rb_2b+m_2Mb_2b+\cdots+aba_nb_n+rb_nb+m_nMb_nb$

and $c_2 = y_1b + y_2b + \cdots + y_nb$, which he sends to Alice,

Step 3: Alice calculates

$$\begin{aligned}
&(c_1 - rc_2) \bmod M = aba_1b_1 + rb_1b + aba_2b_2 + \\
&rb_2b + \cdots + aba_nb_n + rb_nb - \\
&r\left(b_1b + b_2b + \cdots + b_nb\right) = \\
&a\left(a_1b_1 + a_2b_2 + \cdots + a_nb_n\right)b
\end{aligned}$$

Here, Alice uses the two numbers obtained to perform modulo and division operations on the large integer M , resulting in the following three equations:

$$\begin{cases}
c_2 = b_1b + b_2b + \cdots + b_nb \\
c_3 = a\left(a_1b_1 + a_2b_2 + \cdots + a_nb_n\right)b \\
c_4 = m_1b_1b + m_2b_2b + \cdots + m_nb_nb
\end{cases}$$

After eliminating,only the following two equations can be derived:

$$\begin{cases}
c_2a\left(a_1b_1 + a_2b_2 + \cdots + a_nb_n\right) = \\
c_3\left(b_1 + b_2 + \cdots + b_n\right) \\
c_4a\left(a_1b_1 + a_2b_2 + \cdots + a_nb_n\right) = \\
c_3\left(m_1b_1 + m_2b_2 + \cdots + m_nb_n\right)
\end{cases}.$$

Because Bob provides the row vector data

$$\begin{aligned}
&(b_1, b_2, \cdots, b_n) = (-z_{01}\mu_i, z_{01}\lambda_i, \lambda_i\left(y_i - y_{01}\right) - \\
&\mu_i\left(x_i - x_{01}\right), -z_0\mu_i, z_0\lambda_i, \lambda_i\left(y_i - y_0\right) - \mu_i\left(x_i - x_0\right)
\end{aligned}$$

and since there are four parameters λ_i,μ_i,x_i, y_i , only four equations can be obtained when making two determinations on whether the inner product is zero. Although Alice knows the corresponding values of c_2, c_3, c_4 , they are not controlled or given by Alice. Similarly, for the second inner product determination based on the same row vector data provided by Bob, the two equations obtained also involve c_2', c_3', c_4' , not controlled by Alice, making it impossible to obtain equations that only contain pinhole coordinates (x_{01}, y_{01}, z_{01}) without other parameters. Therefore, it is sufficiently secure if each row of Bob's data is only used for two determinations of whether the inner product is zero. Thus, Bob's data is secure. Under such an agreement, due to the effect of modulo operations, Alice's data is obviously secure. Whether $M_2M_1^{-1}(:, 1 : 2)$ changes only requires two inner product determinations for each row of Bob's data. Hence, Theorem 2 corresponding to Theorem 1(4) can determine whether the pinhole coordinate data of both parties are equal.

4 Conclusion

By adopting certain conventions, this paper introduces two theorems derived from the imaging of a pinhole in a planar straight line, specifically focusing on the three-dimensional matrices related to pinhole coordinates. Then, using these theorems, it converts the task of determining whether the pinhole coordinates of both parties are the same into checking whether the vector inner product of the integer components they own is zero. Further, it establishes a protocol for whether this vector inner product is zero and analyses the potential for information leakage from a theoretical perspective. Several improvement plans are proposed for various scenarios. Apart from the dynamic encryption involved in the initial data conversion to pinhole coordinates, the operations normally involve only integer addition, subtraction, and multiplication. The final supplementary protocol for rigorously determining equality involves only large prime numbers and modulo operations, avoiding the inaccuracies of decimal operations and the inefficiency of high-power computations. Thus, our protocol is highly practical and efficient.

For addition, subtraction, and multiplication of large integers that exceed the default computational capacity of computers, integers can be treated as vectors with each digit as a component. By customizing the processing of vectors and combining parallel algorithms, the multiplication and addition/subtraction of large-digit integers can be calculated. This approach extends the range of operations and greatly enhances computational efficiency, allowing this paper to determine whether large-digit data are equal.

If the data to be compared involve practical tasks like text proofreading and intersection of sets, frequent comparisons of whether strings are equal are required. Determining if strings are identical through binary conversion can be understood as determining if data are equal. An advantage of this paper is that each time the agreed-upon text strings to be compared are equal in length, longer than in other literature, allowing for an efficient protocol without the need for modulo operations. The equality judgement of data between the two-party is innovatively converted into judging whether the inner product of the vectors owned by the two parties is zero. Based on the specific property of these vectors, a security protocol is established to determine whether the inner product is zero, which realizes the protocol scheme of judging the data equality of two-party under the semi-honest model and analyses its security theoretically. If we allow one party can access the other's same data two or more times, this data may not safe, so we need the key stream for dynamic data encoding such that the data of two parties are encoded into different ciphertext each time they are judged to be equal. As high-power integer operations are not needed, this results in a relatively highly efficient and secure protocol. If the common factor is easily detected by Bob, Alice and Bob could only compute the matrix $M_2M_1^{-1}(:,1)$ two or more times. A large number of practical simulations show that there is no such thing as the original unequal information judgement of equality.

This scheme may not fit for malicious model, we could use the existing agreements [18]- [19] of malicious model to determine whether those inner products are zeroes, by the conversion in this paper we also could complete the equality judgement.

Acknowledgment

The authors would like to thank Dr. Yi Deng of Institute of Information Engineering, Chinese Academy of Sciences, for helpful discussion during this research. The research of C. Gui is supported by University of Macau UMDF Professorial Fellowship and The Science and Technology Development Fund, Macao S.A.R (FDCT) Project 0024/2023/RIB1.

References

[1] Yao, A. C. Protocols for secure computations. Proc. of the 23rd Annual IEEE Symposium on Foundations of Computer Science, 1982.

[2] Cleve.R. Limits on the security of coin flips when half the processors are faulty(extended abstract) Proceedings of the 18th Annual ACM symposium on Theory of Computing (STOC), 364–369, ACM Press, 1986.

[3] Goldwasser, S. How to play any mental game, or a completeness theorem for protocols with an honest majority. Proc. the Nineteenth Annual ACM STOC'87., 1987.

[4] Lindell, Y. , Pinkas, B. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In Proceedings of the 26th annual international conference on Advances in Cryptology, EUROCRYPT '07, pages 52–78, Berlin, Heidelberg, 2007. Springer-Verlag.

[5] Lindell, Y., Pinkas, B. Secure Multiparty Computation for Privacy-Preserving Data Mining. Journal of Privacy and Confidentiality, 1(1), 59–98. 2008.

[6] Ishai, Y. , Prabhakaran, M. , Sahai, A. Founding Cryptography on Oblivious Transfer – Efficiently. Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. DBLP.

[7] Hazay, C., Lindell, Y. Efficient Secure Two-Party Protocols: Techniques and Constructions. Springer. 2010.

[8] Lindell, Y. , Pinkas, B. Secure two-party computation via cut-and-choose oblivious transfer. Journal of Cryptology, 25(4), 680-722. 2012.

[9] Gordon, S. D. , Katz, J. Partial fairness in secure two-party computation. Journal of Cryptology, 25(1), 14-40. 2012.

[10] Boudot, F. , Schoenmakers, B. , Jacques Traoré. A fair and efficient solution to the socialist millionaires' problem. 2001.

[11] Lo, H. K. Insecurity of quantum secure computations. Physical Review A, 56(2), 1154-1162. 1998.

[12] Bogdanov, D., Laur, S., Willemson, J. Sharemind: A framework for fast privacy-preserving computations. Journal of Computer Security, 2008.

[13] Damgård, I. , Fehr, S. , Lunemann, C. , Salvail, L. , Schaffner, C. Improving the security of quantum protocols via commit-and-open. Springer-Verlag.2009.

[14] YuGuang, Y. , QiaoYan, W. An efficient two-party quantum private comparison protocol with decoy photons and two-photon entanglement. Journal of Physics A Mathematical & Theoretical, 43(20), 209801. 2010.

[15] Chen, X. B. , Xu, G. , Niu, X. X. , Wen, Q. Y. , Yang, Y. X. An efficient protocol for the private comparison of equal information based on the triplet entangled state and single-particle measurement. Optics Communications, 283(7), 1561-1565. 2010.

[16] Fu, Z., Ren, K., Shu, J., Sun, X., Huang, F. Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement. IEEE Transactions on Parallel and Distributed Systems, 27(9), 2546-2559. 2015.

[17] Couteau, G. New Protocols for Secure Equality Test and Comparison. Applied Cryptography and Network Security. Springer, Cham. 2018.

[18] Wenliang Du. A study of several specific secure two-party computation problems. PhDthesis, Purdue University, West-Lafayette, Indiana, 2001.

[19] Rafael Dowsley, Jeroen Graaf, Davidson Marques, Anderson C. A. Nascimento.A Two-Party Protocol with Trusted Initializer for Computing the Inner Product.Information Security Applications, 6513,337-350,2011