Research Article

Secure Two-Party Distance Computation Protocols with a Semihonest Third Party and Randomization for Privacy Protection in Wireless Sensor Networks

Haiping Huang, Tianhe Gong, Ping Chen, Guoxia Qiu, and Ruchuan Wang

1College of Computer, Nanjing University of Posts and Telecommunications, Mail Box 157, No. 66, Xinmofan Road, Nanjing 210003, China
2Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China
3Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing 210003, China
4College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Correspondence should be addressed to Haiping Huang; hhp@njupt.edu.cn

Received 24 July 2014; Accepted 23 November 2014

Academic Editor: Muhammad Khurram Khan

Copyright © 2015 Haiping Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Scenarios in which two nodes who distrust each other in wireless sensor networks (WSNs) would like to know the distance between them are considered. The scenario is designed to protect the private information of WSNs, in this case each node’s location, from the other nodes and from a passive attacker. The goal of the present work is to provide two novel and secure two-party distance computation protocols based on a semihonest model, the first with aid of a third party and the second based on randomization technique. Both of these protocols can extend the calculated value into a real number field. The output of the distance computation and the intermediate values in the proposed protocols are also private and not accessible to a third party or any other attackers. When executing these two protocols, security is guaranteed, and the performances of communication and computation of them are found to be satisfactory when compared to those of other similar protocols.

1. Motivation

With the development of the computer and communication technologies, wireless sensor networks have gradually extended into the fabric of human society. Communications between people and things have become a reality so that many researchers have begun to pay attention to wireless sensor networks, where many techniques require secure authentication and secure privacy computation. This is because privacy protection is a significant research point which cannot be ignored in today’s wireless sensor applications. The privacy of wireless sensor networks’ entity may appear to be either a natural property or a social characteristic depending on the requirements of specific scenarios. The matter of defining privacy in terms of different applications and realizing privacy protection can be explained in detail through the following scenarios regarding secure communications.

Wireless sensor networks can usually be deployed in the enemies’ battle field to monitor military information. Once found, each node has the risk of being captured by enemies so that all nodes do not trust each other. However, these nodes have to cooperate with each other in this unreliable environment in order to complete tasks of data collection. For example, Node A decides to expand its coverage area through enhancing transmitting power in one region to obtain more sensing data. Node A also notices that Node B, possibly the adjacent node of Node A, is also expanding the coverage area of his located region. But the two nodes do not want to collect data in the same area, so they want to know whether there are overlapping areas between their respective regions without disclosing the corresponding location information, in order to avoid the danger of being captured caused by the leakage of location information. If Node A and Node B are able to calculate the distance between their respective regions in the case of keeping geographical location information...
confidentiality and meanwhile telling each other their respective coverage radiiuses, then this problem can be effectively solved: if the distance is too close, it indicates that there is overlap; otherwise there is no overlap.

Location information becomes important and confidential in the above scenarios, though the utilization of the respective positions of two participants is the most common method for calculating Euclidean distance. In this case, secure two-party distance computation without revealing location information becomes a vital problem.

Secure two-party computation, introduced by Yao, allows two parties to jointly compute any function using their inputs in such a way that (1) the output of the computation is correct and public and (2) the inputs are not revealed to both users [3]. Secure calculation of the distance between two parties (or two nodes in WSNs) is a specific case of secure two-party computation, and it provides a solution to the scenario described above. Currently, there exist two ways of computing the distance, and both of them allow the location of each party to be kept secret. The first requires the help of a third party, and the other does not. The primary contribution of this paper is that it puts forward two novel, secure, and efficient protocols that can be used in accordance with the above two methods.

The rest of the paper is organized as follows. The next section reviews related works in the field of secure two-party distance computation. Section 3 presents the preliminaries and assumptions. Section 4 provides two novel and secure two-party distance computation protocols used in WSN, the first involving a semihonest third party and the other involving randomization technique, and their respective correctness proofs. Security analysis and performance comparisons between the proposals and previous work are presented in Sections 5 and 6, respectively. Further discussions about sensitive issues are given in Section 7. Section 8 concludes this paper.

2. Related Work

Secure two-party computation is widely used in network security protocol design to protect important information related to location, identity, and healthcare. Recently, location information during the use of location-based services (LBSs) has raised considerable concerns [12], especially in distance calculation application. Currently, secure two-party distance protocols are generally divided into three categories. The first [4–7] is established using a homomorphic encryption mechanism that serves as the secure basis of privacy protection. The second [8, 13] is established with the assistance of a third party, usually a semihonest third party. The third [14] is established with the randomization technique.

As the first type of protocol, [4] describes and analyzes a scheme for biometric authentication and it uses homomorphic encryption for secure calculation of Hamming Distance without revealing the participants’ secret information. The work [5] puts forward a secure two-party computation protocol of Euclidean distance using Paillier homomorphic encryption [11] and this protocol is implemented for private querying of face images and maintains low communication overhead. The work [6] also proposes a secure multiparty distance computation protocol, which is also designed based on Paillier homomorphic encryption without a third party. The work [7] suggests a secure, privacy-preserving opportunistic computing (SPOC) framework for healthcare emergency based on a secure two-party product protocol that can be used to design a secure two-party distance computation protocol. Pronounced computational complexity is the common weakness of secure two-party computation protocols based on homomorphic encryption mechanisms. This is difficult to use with resource-constrained outdoor wireless link networks.

In the second category protocol, two participants are unable to calculate the distance between them if there is no third party involved. The protocol in [13] involves a quantum private comparison based on the presence of a semihonest third party and it enables two parties to determine whether their information matches without revealing the specifics. The work [8] allows the participant to calculate the distance based on an honest third party for comparison of local data and simultaneously causes private values to be fully grasped by the third party, which is followed by obvious information leakage and security issues that cannot be ignored. Although intervention by a third party may improve the efficiency of computation and communication, this increases network costs. This issue of decreasing the number of secrets learned by a third party is significant here. The distinction between honest and semihonest third parties is illustrated in Section 3.2.

Broad application of homomorphic encryption mechanisms results in less use of other digital disguise techniques, such as randomization methods, which can be used to design secure two-party computation protocols. Randomized disguises obscure the real data by adding random elements to them to facilitate privacy protection. For example, [14] provides a randomization method by performing a linear transformation on the real data or a random permutation replacement of the real data to achieve a secure two-party distance computation protocol. The protocol in [9] also involves randomization method and 1 out of m oblivious transfer protocol [11]. Designing a secure and efficient two-party distance computation protocol based on randomized masquerade is mathematically difficult. Many current protocols, such as one described in a previous study [8], can only be used in integer domains and cannot be expanded into real number domains. This limits their use.

In wireless sensor networks, literatures related to privacy protection of location and distance [15] can be traced back to 2007. Actually only few literatures apply the distance computation protocols based on privacy protection to data security of wireless sensor networks. To the best of our knowledge, [16] is the first paper that uses the secure distance calculation protocol to solve location privacy in WSNs. The work [17] presents an overview of the solutions that provide source location privacy such as anonymity and unobservability, within a WSN, in relation to the assumptions about the adversary’s capabilities. A location privacy routing protocol (LPR) is proposed in [18] to achieve path diversity, and, combining with fake packet injection, LPR is able to
minimize the traffic direction information that an adversary can retrieve from eavesdropping. However, LPR does not provide the data-level privacy protection. The work [19] found out that Lu et al.'s protocol in [7] still has some secure flaws such as user anonymity and mutual authentication, and it presents an improved mobile-healthcare emergency system based on extended chaotic maps for applications of wireless body sensor networks (BSNs).

3. Preliminaries and Premises

3.1. Definition of Secure Two-Party Computation. There are two participants in WSNs, Node A and Node B. Node A selects a secret input \( x \) and Node B selects another one \( y \). They hope to calculate the function \( f(x, y) \) and produce final results but they must keep their respective inputs secret. We call this computation process “secure two-party computation.” Here, \( x \) (or \( y \)) may be the location information of Node A (or Node B), and \( f(x, y) \) may be the distance between the two nodes.

3.2. Semihonest Model. Secure two-party computation should guarantee the safety of the information even if one participant behaves deceitfully in the execution of the protocol. The attacker may perform one of two types of attacks: one is active that is called “malicious (or adversarial) model [20]” and another is passive, here called the “semihonest model.” In the malicious model, the attacker manipulates one of the participants into executing the protocol improperly. For example, the attacker may cause a participant to substitute mendacious data for real input in order to interrupt implementation of the protocol. In the semihonest model, the attacker only grasps the entirety of the information of the captive participant, including those intermediate data acquired from the other participant, and the purpose of the attacker is solely to reveal some private information, but the protocol is carried out correctly and all inputs are true. Some researchers refer to the semihonest participant as “honest but curious” [21].

Based on the above, the reliability of a third party can be defined (hereinafter abbreviated “TP”). There exist three cases. In the first case, the TP is honest, and both the participants send their secrets to the TP without worrying that the TP will leak any private information. This case is perfect but impractical. In the second case, the TP is dishonest or malicious, and neither participant believes the TP. This case is dangerous. In the third case, the TP is semihonest. The TP executes the protocol faithfully, and all kinds of intermediary data and computational results are recorded. The TP tries to make this information publicly available. In WSNs, base station or the cluster-header usually acts as the role of TP.

In the present paper, it is supposed that both participants and TP all follow the semihonest model. Generally, the semihonest secure two-party computation model can be illustrated in detail as follows. This can be used in security proofs of protocols.

There exists a mapping function \( f : \{0, 1\}^* \times \{0, 1\}^* \mapsto \{0, 1\}^* \times \{0, 1\}^* \), where \( f_1(x, y) \) is the first element and \( \pi \) denotes the secure two-party computation protocol. During the implementation of \( \pi \), the view of the first party \( \text{view}_1^n(x, y) \) is represented by \( (x, r_1^1, m_1^1, m_2^1, \ldots, m_i^1) \) (analogously \( (y, r_2^1, m_1^2, m_2^2, \ldots, m_i^2) \) denotes the view of the second party \( \text{view}_2^n(x, y) \)). Here, \( x \) is the input, \( r_1^1 \) denotes the result of random coin toss, and \( m_i^1 \) stands for the \( i \)th intermediate message received by the first party. When \( \pi \) is complete, the output of the first party is marked with \( \text{output}_1^n(x, y) \) (the output of the second party is \( \text{output}_2^n(x, y) \)) and the output of \( \pi \) is \( \text{output}_n^n(x, y) = (\text{output}_1^n(x, y), \text{output}_2^n(x, y)) \).

Generally, if there exists a probabilistic polynomial-time algorithm (denoted by \( S_1 \) or \( S_2 \)) to execute the privacy computation of \( f \), it can be concluded that the following is true:

\[
\left\{ (\text{view}_1^n(x, y), \text{output}_1^n(x, y)) \right\}_{x,y} = \left\{ \left( (f_1(x, y), S_2(y, f_2(x, y))) \right) \right\}_{x,y} \equiv \left\{ (\text{output}_1^n(x, y), \text{view}_2^n(x, y)) \right\}_{x,y}.
\]

Here, the meaning of “\( \equiv \)” is “calculation of indiscernibility,” \( \text{output}_n^n(x, y) \) is entirely determined by \( \text{view}_n^n(x, y) \), and both of them are stochastic variables. \( S_1 \) and \( S_2 \) are called the time simulators.

3.3. Definition of Distance. There are many definitions of distance between two participants, including the Minkowski distance, Hamming distance, and Levenshtein distance [22]. The Minkowski distance is the most common one. It can be expressed using

\[
d(p) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}.
\]

Here \( x_i \) and \( y_i \) are from \( P = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \) and \( Q = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n \), respectively. When \( p = 1 \), it becomes the Manhattan distance; when \( p = \infty \), it becomes the Chebyshev distance; when \( p = 2 \), it becomes the Euclidean distance, which can be described by

\[
d(2) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}.
\]

In this paper, the Euclidean distance between two participants is calculated exactly in the two proposed secure protocols.

3.4. Privacy Measure. Most researchers use the “calculation of indiscernibility” method to realize security proofs, but this method cannot quantify privacy. If it is necessary to prove that one protocol is absolutely secret (perfect zero leakage), privacy measures become essential because it can validate safety of protocols against attackers more efficiently. However, currently there is no common method of measuring privacy.

\[\text{privacy}(\pi) = \frac{\text{indiscernibility}(\pi)}{\text{information}(\pi)}\]

where \( \text{indiscernibility}(\pi) \) represents the indiscernibility of the protocol, and \( \text{information}(\pi) \) represents the information content of the protocol.
Although the utilization of information theory can quantify privacy, it has limitations [23, 24]. Specifically, it is only available for the protocols whose input has a linear relationship with output. It cannot be applied to nonlinear protocols.

Several symbols and concepts regarding the measurement of privacy are given below. These were used to confirm the validity of the proposed protocol.

\(X_i^n\) denotes the input of the \(i\)th participant in \(\pi\), and \(msg_j^n\) is described as the message received by the \(i\)th participant. After the beginning of \(\pi\), each participant receives some information from his partner. In this way, the following can be defined:

- \(H_i^T = H(X_i^n)\): the entropy of stochastic variable \(X_i^n\);
- \(H_j^n = H(X_j^n \mid msg_j^n)\): the conditional entropy of \(X_j^n\) when \(msg_j^n\) is given and when it implies the uncertainty on \(X_j^n\) of the \(j\)th participant while receiving \(msg_j^n\);
- \(\min_{j,i}(H_j^n \mid H_i^T)\): the relative privacy of \(\pi\);
- \(\min_{j,i}(H_i^T)\): the absolute privacy of \(\pi\).

Relative privacy is believed to be more likely to reflect the degree of privacy divulgence than the absolute privacy, and it can be standardized to a value between 0 and 1 where 0 represents full leakage and 1 indicates perfect safety (perfect zero leakage).

**Theorem 1** (see [23]). Consider that \(V, C \in GF(p)^n\) (both \(V\) and \(C\) are one-dimensional vectors); \(A\) is a matrix with \(m\) rows and \(n\) columns whose element belongs to \(GF(p)^n\) and \(\text{rank}(A) = k\). If there exists a functional dependency relationship among \(A, V, \) and \(C\), which can be expressed by \(A \ast V = C\), and meanwhile if \(V\) has \(n\) unknown quantities, then

\[ H(V \mid C) = (n - k) \log p. \]

The proof for Theorem 1 is shown in a previous study [23].

### 4. Two Novel Secure Two-Party Distance Computation Protocols

#### 4.1. Description of the TPEDP Protocol

In this section, a novel third-party-based Euclidean distance protocol (abbreviated TPEDP) is used to securely calculate the distance between two participants of Node A and Node B. Based on semihonest model, the TPEDP is divided into three phases: the input phase, computation phase, and output phase.

**Input Phase.** Node A selects a secret vector \(P = (x_1, x_2, \ldots, x_n)\), and Node B chooses another one \(Q = (y_1, y_2, \ldots, y_n)\), where \(n \geq 1\) and their location information are represented by \(P\) and \(Q\), respectively.

**Output Phase.** Through two-party calculation, \(f(P, Q)\) is the final result that denotes Euclidean distance \(|PQ|^2\). Node A grasps \(f_a\) and Node B keeps \(f_b\). Equation (4) shows the relationship among those symbols:

\[ f_a = f(P, Q) + f_b \]  

(4)

At last, Node A declares \(f_a\) and Node B reveals \(f_b\), and they share the resulting \(f(P, Q)\) according to (4).

**Computation Phase**

**Step 0.** The TP (a third party, i.e., a base station or a cluster-header) randomly generates two vectors, \(Ra = (ra_1, ra_2, \ldots, ra_n)\), \(Rb = (rb_1, rb_2, \ldots, rb_n)\), and two numbers, \(p\) and \(q\). It causes them to satisfy (5). Then the TP sends Ra and p to Node A and Rb and q to Node B. Consider

\[ (Ra - Rb) \cdot (Ra - Rb)^T = p + q. \]  

(5)

**Step 1.** Node A receives Ra, calculates \(P_1 = P + Ra = (x_1 + ra_1, x_2 + ra_2, \ldots, x_n + ra_n)\), and sends \(P_1\) to Node B.

**Step 2.** Node B computes \(Q_1 = Rb - Q = (rb_1 - y_1, rb_2 - y_2, \ldots, rb_n - y_n)\) using the received vector Rb and its secret Q. Then Node B generates a random number \(v\), together with \(P_1\), and establishes (6). Finally, Node B sends \(Q_1\) and \(s\) to Node A. Consider

\[ s = (P_1 + Q_1) \cdot (P_1 + Q_1)^T + 2Rb \cdot Q^T \]

- \(2Rb \cdot P_1^T - q + v. \)  

(6)

**Step 3.** After receiving \(Q_1\) and \(s\), Node A calculates \(u\) in terms of

\[ u = s - 2Ra \cdot Q_1^T - 2Ra \cdot P_1^T - p. \]  

(7)

Finally, Node A and Node B determine the distance \(|PQ|^2\) (i.e., \(d^2\)) with respect to

\[ u = d^2 + v. \]  

(8)

Here, \(f_a = -u\), \(f_b = -v\), and \(f(P, Q) = d^2\) satisfy the form of (4).

**Proof (correctness proof).** The TPEDP can achieve the correct result of Euclidean distance between two participants.

**Proof.** The arithmetic expression of (6) can be expanded as follows:

\[ s = (P_1 + Q_1) \cdot (P_1 + Q_1)^T + 2Rb \cdot Q^T \]

- \(2Rb \cdot P_1^T - q + v\)

= \sum_{i=1}^{n} (x_i + ra_i + rb_i - y_i)^2 + 2 \sum_{i=1}^{n} rb_i y_i

- \(2 \sum_{i=1}^{n} rb_i (x_i + ra_i) - q + v\)

= \sum_{i=1}^{n} ((x_i - y_i) + (ra + rb_i))^2 + 2 \sum_{i=1}^{n} rb_i y_i

- \(2 \sum_{i=1}^{n} rb_i (x_i + ra_i) - q + v\)

\[ = d^2 + v. \]
From (5), the quality of some outdoor wireless sensor network environments do not allow the existence of a reliable third party, so the two nodes must complete security distance calculation relying only on each other. A randomization technique is used to avoid the high computational complexity produced by homomorphic encryption, and the design of an efficient two-party protocol is described in this section. It is here called the randomization-based Euclidean distance protocol (abbreviated REDP). The difficulty of designing REDP is that it requires randomly camouflaging vectors in the domain of real numbers and it involves exchanging some elements within those modified vectors.

It can be supposed that Node A and Node B are semihonest participants and operations in input and output phases of the REDP are identical to those of the TPEDP. Then only the steps of computation phase are displayed as follows.

**Computation Phase**

**Step 1.** Node A generates a random sequence \( \{ra_0, ra_1, \ldots, ra_n\} \) where \( ra_i \neq 0 \), \( i \in [0, n] \), and, for each \( x_i \) in \( P = (x_1, x_2, \ldots, x_n) \), Node A calculates \( x'_i = ra_i x_i + ir a_i \) (\( i = 1, 2, \ldots, n \)) which creates a new vector \( P_1 = (x'_1, x'_2, \ldots, x'_n) \). Node A continues to calculate the values of \( \lambda \) and \( x''_i \) according to (11) and (12), respectively, where \( u, e, \) and \( w (u, e, w \neq 0) \) are random numbers chosen by Node A:

\[
\lambda = \frac{1}{2} ra_0 \left( u + \sum_{i=1}^{n} x'_i \right),
\]

\[
x''_i = \frac{w - ra_i}{e}, \quad (i = 1, 2, \ldots, n).
\]

All \( x''_i \) form a new vector \( P_2 = (x''_1, x''_2, \ldots, x''_n) \), and Node A sends \( P_1, P_2, \) and \( \lambda \) to Node B but keeps \( \{ra_0, ra_1, \ldots, ra_n\}, e, w, \) and \( u \) secret.

**Step 2.** Node B constructs two fresh vectors \( Q_1 = (y'_1, y'_2, \ldots, y'_n) = (ra_0 y_1, ra_1 y_2, \ldots, ra_n y_n) \) and \( Q_2 = (y''_1, y''_2, \ldots, y''_n) = (ra_0 y_1, 2ra_2 y_2, \ldots, 2ra_n y_n) \) with a random number \( r_b \) (\( r_b \neq 0 \)) and \( Q (Q = (y_1, y_2, \ldots, y_n)) \). Then Node B computes \( h_1, h_2, \) and \( h_3 \) in terms of (13), (14), and (15), separately, and \( P_1, P_2, \) and \( \lambda \) are obtained from Node A. Consider

\[
h_1 = \lambda r_b + P_1 * Q_1 = ra_0 r_b \left( u + \sum_{i=1}^{n} x'_i \right) + \sum_{i=1}^{n} x'_i y'_i
\]

\[
= \frac{1}{2} ra_0 r_b \left( u + \sum_{i=1}^{n} x'_i \right) + \sum_{i=1}^{n} \left( ra_0 x_i + ir a_i \right) r_b y_i,
\]

\[
h_2 = P_2 * Q_2 = \sum_{i=1}^{n} x''_i y''_i = \sum_{i=1}^{n} \left( w - \frac{ra_i}{e} \right) r_b y_i,
\]

\[
h_3 = \sum_{i=1}^{n} y''_i.
\]

At last, Node B sends \( h_1, h_2, \) and \( h_3 \) to Node A but keeps \( r_b \) secret.

**Step 3.** Node A determines the value of \( m \) on the basis of (16) and sends it to Node B:

\[
m = \frac{(h_1 + eh_2 - wh_3)}{ra_0}.
\]

**Step 4.** Node B obtains the value of \( v \) according to

\[
v = \frac{2m}{r_b} + \sum_{i=1}^{n} y''_i.
\]

Finally, Node A and Node B share the value of \( |PQ|^2 \) (i.e., \( d^2 \)) in the light of

\[
v = d^2 + u.
\]

Here \( f_u = u, f_b = v, \) and \( f(P, Q) = d^2 \) all satisfy the form of (4).

\[
\begin{align*}
&= \sum_{i=1}^{n} (x_i - y_i)^2 + \sum_{i=1}^{n} (ra_i - rb_i)^2 + 4 \sum_{i=1}^{n} ra_i rb_i \\
&\quad + 2 \sum_{i=1}^{n} (x_i - y_i) (ra_i + rb_i) \\
&\quad + 2 \sum_{i=1}^{n} rb_i y_i - 2 \sum_{i=1}^{n} rb_i (x_i + ra_i) - q + v \\
&= \sum_{i=1}^{n} (x_i - y_i)^2 + \sum_{i=1}^{n} (ra_i - rb_i)^2 + 2 \sum_{i=1}^{n} x_i ra_i \\
&\quad + 2 \sum_{i=1}^{n} ra_i (rb_i - y_i) - q + v \\
&= \sum_{i=1}^{n} (x_i - y_i)^2 + \sum_{i=1}^{n} (ra_i - rb_i)^2 + 2 Ra * P^T \\
&\quad + 2 Ra * Q_1^T - q + v.
\end{align*}
\]

From (5), the equality of \( p + q = \sum_{i=1}^{n} (ra_i - rb_i)^2 \) can be deduced. It can be substituted into (10). Finally, the expression of \( u = \sum_{i=1}^{n} (x_i - y_i)^2 + v = |PQ|^2 + v \) can be concluded and the validity of the TPEDP is proven. 

4.2. Description of the REDP Protocol. Some outdoor wireless sensor network environments do not allow the existence of a reliable third party, so the two nodes must complete security distance calculation relying only on each other. A randomization technique is used to avoid the high computational complexity produced by homomorphic encryption, and the design of an efficient two-party protocol is described in this section. It is here called the randomization-based Euclidean distance protocol (abbreviated REDP). The difficulty of designing REDP is that it requires randomly camouflaging vectors in the domain of real numbers and it involves exchanging some elements within those modified vectors.
Proof (correctness proof). According to (16), (17), and (18), (19) can be deduced:

\[
d^2 = v - u = \frac{2m}{r_b} + \sum_{i=1}^{n} y_i^2 - u
\]

It makes use of the expressions of (13), (14), and (15) to displace \(h_1, h_2, \) and \(h_3\) in (19), respectively. Then the following can be concluded:

\[
d^2 = -\left[\frac{1}{2} \right] ra_0 r_b \left( u + \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} r_q r_b x_i y_i \right)
\]

\[
+ \sum_{i=1}^{n} y_i^2 - u
\]

\[
= u + \sum_{i=1}^{n} x_i^2 - 2 \sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 - u
\]

\[
= \sum_{i=1}^{n} (x_i - y_i)^2.
\]

Equation (20) shows the exact computation of the expression of Euclidean distance between Node A and Node B. In this way, the proof is complete. \(\square\)

5. Security Analysis

5.1. Security of the TPEDP Protocol. In this section, the security of the TPEDP protocol is proven. Node A and Node B cannot learn any private vectors from each other except for the distance between them. This provides a detailed analysis of each step of the TPEDP.

1. In Step 0, the TP receives no reply from either Node A or Node B, so it cannot gain any information from either of the two participants.

2. In Step 1, Node B learns \(P_1 = P + Ra\) from Node A but cannot analyze \(P\) because Node B knows nothing about \(Ra\). After receiving \(P_1\), Node B encounters a group of nonhomogeneous linear equations with \(2n\) unknown quantities but only \(n\) equations. These equations cannot be solved in an algebraic field.

3. In Step 2, Node A receives \(s\), which is the result of scalar products for \(P_1, Q_1\). For Node A, \(s\) and \(Q_1\) are equivalent to a group of nonhomogeneous linear equations, including \(2n + 2(Q, Rb, q, v)\) unknown quantities, but the \(n + 1\) equations are not enough to solve for those elements of \(Q_1\).

4. In Step 3, owing to adding stochastic number \(v\) into (6) by Node B, Node A cannot hold the final result by itself. It means that \(u\) is only available to Node A and \(v\) is only available to Node B. For any one participant, (8) has two unknown quantities and an infinity of solutions. In this way, neither Node A nor Node B can deduce the value of \(d^2\) from their respective data alone. So the scenario protects the secrecy of the information regarding the location of the two participants.

In summary, none of the information is compromised during communications among Node A, Node B, and TP. In order to render the procedure of security analysis more convincing, the calculation of indiscernibility method was adopted for further validation.

Theorem 2. The TPEDP (marked with \(\pi\)) ensures that Node A and Node B cannot acquire private input of each other.

Proof. Because both Node A and Node B are semihonest participants, it is necessary to prove, in the TPEDP, that the intermediate information obtained by the passive attacker cannot be differentiated from those produced in no-attack and real situations, that is, calculation of indiscernibility.

A simulator must be constructed to imitate all of information acquired by a passive attacker in the executing process of the TPEDP. It can be discretionarily determined whether Node A or Node B is captured by the attacker. For this reason, the simulator \(S_1\) can simulate data from the view of Node A view\((P = (x_1, x_2, \ldots, x_n)\), \(Q = (y_1, y_2, \ldots, y_n)\)). In this way, this proof process can be used to verify whether \(\{S_1(P, f(P, Q), f(Q, P))\}\) and \(\{view^{\pi}_{S_1}(P, Q), output^{\pi}_{S_1}(P, Q)\}\) are equivalent, where view\((P, Q) = (P, r^1, m^1, m^2, \ldots, m^1)\) and location information \(P = (x_1, x_2, \ldots, x_n)\), and the distance \(d\) serve as the inputs of \(S_1\).
In Step 3, $S_1$ incites Node A to calculate the simulant parameter $u' = s' - 2Ra_i \ast s_i'Q_i - 2Ra_i \ast P_i - p_i$.

Output Phase of $\pi_i$. Finally, $f'_a = -u'$, $f'_b = -v'$, and all satisfy the form of (4); that is, $f'_b = f(P, Q) + f'_a$. After the exchange of data between Node A and Node B, through simulation, $S_1$ obtains all information of Node A: $\{f'_a, f'_b, f(P, Q)\}$.

However, in each of the steps outlined above, all of the data (denoted by $S_j(P, d) = \{P, r^', s', Q_i, d_i\}$) handled by $S_1$ cannot be distinguished from those produced in the normal execution of $\pi_i$. Because the view of Node A is $\text{view}\_i^{\pi_i}(P, Q) = \{P, r^', s, Q_i\}$ and the output of $\pi_i$ is $\text{output}\_i^{\pi_i}(P, Q) = \text{output}\_i^{\pi_i}(P, Q) = d$, $\{S_j(P, d), d\} \equiv \{\text{view}\_i^{\pi_i}(P, Q), \text{output}\_i^{\pi_i}(P, Q)\}$ can be concluded. This indicates “calculation of indiscernibility.”

Likewise, if the attacker suborns Node B, then the attacker will gather all information in Node B’s possession using another simulator, $S_2$. The view of Node B is $\text{view}_{2i}^{\pi_i}(P = (x_1, x_2, \ldots, x_n), Q = (y_1, y_2, \ldots, y_n))$, and the same conclusion of “calculation of indiscernibility” can be drawn: $\{f_i(P, Q), S_i(Q, f_i(P, Q))\}$ and $\{\text{view}_{2i}^{\pi_i}(P, Q), \text{output}_{2i}^{\pi_i}(P, Q)\}$ are equivalent.

If the attacker captures the TP, then the attacker cannot realize the simulation because the TP has not obtained any information from Node A or Node B at any point in the protocol.

This proves Theorem 2. □

5.2. Security of the REDP Protocol. Node A and Node B cannot obtain any private vectors from each other except for the distance value between them. The following is a detailed analysis of each step in the REDP process.

(1) In Step 1, Node B receives some data from Node A, including $P_1 = (x_1^1, x_2^1, \ldots, x_n^1)$, $P_2 = (x_1^2, x_2^2, \ldots, x_n^2)$, where $x_i^1 = ra_i x_i + ir_{a_i} x_i = (w - r_{a_i})/e$ ($i = 1, 2, \ldots, n$). However, because Node B cannot learn the specific values of $ra_i$, $w$, or $e$, it is faced with the difficulty of solving $x_i$ with $2n$ nonlinear polynomial equations but $2n + 3$ unknown quantities. These can be expressed by (22), where $b_i\ (j = 1, 2, \ldots, n, n + 1, \ldots, 2n)$ is a specific constant. Consider

\[
\begin{align*}
x_{2n+1}x_1 + x_1 &= b_1, \\
x_{2n+2}x_1 + ix_1 &= b_1, \\
&\vdots \\
x_{2n+1}x_n + nx_1 &= b_n, \\
x_{2n+2} - x_{2n+1} &= b_{n+1}, \\
&\vdots \\
x_{2n+3} - x_{2n+2} &= b_{n+3}, \\
&\vdots
\end{align*}
\]

Equation (22) has infinity of solutions. In some cases, such as $ra_q = 1, e = 1$, (22) can be degraded to a group of nonhomogeneous linear equations group represented by (23). Here, $b_j\ (j = 1, 2, \ldots, n, n + 1, \ldots, 2n)$ is a specific constant. Consider

\[
\begin{align*}
x_1 + x_{n+1} &= b_1, \\
&\vdots \\
x_{2n+1} - x_{n+1} &= b_{n+1}, \\
&\vdots \\
x_{2n+2} - x_{n+2} &= b_{n+3}, \\
&\vdots
\end{align*}
\]

Because (23) can be solved using features of the nonhomogeneous linear equations, it shows that the coefficient matrix and the augmented matrix of the equations outlined above have the same rank. For this reason, only the rank of the coefficient matrix is considered. This is displayed as follows:

\[
A = \begin{pmatrix}
1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 & 2 & 0 & \cdots & 0 & 0 \\
& \vdots & & & & \vdots & & & & & \vdots \\
0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 & n & 0 \\
0 & 0 & \cdots & 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 1 \\
0 & 0 & \cdots & 0 & 0 & 0 & -1 & 0 & \cdots & 0 & 1 \\
& \vdots & & & & \vdots & & & & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 1 \\
\end{pmatrix}
\]

\[r(A) = 2n.\]
equations) is less than the number of unknown quantities. Even in some extreme situations, (22) still has an infinity of solutions. Node B cannot discriminate \( u \) or \( r_0 \) from the value of \( \lambda \). In this way, the obtained data by the REDP are not sufficient for detection of any private information regarding location vector \( \mathbf{P} = (x_1, x_2, \ldots, x_n) \), which is owned by Node A.

(2) In Step 2, Node A encounters the same problem with respect to solving a group of nonlinear equations. In extreme situations (e.g., if Node B selects \( r_0 = 1 \)), Node A may choose a zero vector \( \mathbf{P} = 0 \) in input phase. This can be used repeatedly in (13) and (14) after Node A receives \( h_1, h_2, \) and \( h_3 \). Finally, Node A can deduce the value of \( r_0 \) with considerable probability of success. Even so, Node A still needs to solve a group of nonhomogeneous linear equations with \( n + 1 \) unknown quantities and only three equations. This problem has an infinite number of solutions. Just by virtue of \( r_0 \), Node A is unable to infer Node B’s confidential location information \( \mathbf{Q} \).

(3) In Step 3, if Node B utilizes the expressions of (13), (14), and (15) to successively substitute for the values of \( h_1, h_2, \) and \( h_3 \) in (16), Node B can produce
\[
m = \frac{- (h_1 + \varepsilon h_2 - w h_3)}{r_0} \cdot \frac{1}{2} r_0 \left( u + \sum_{i=1}^{n} x_i^2 \right) - r_0 \sum_{i=1}^{n} x_i y_i.
\]

There is a variety of ways in which Node B might separate individual components such as \( u + \sum_{i=1}^{n} x_i^2 \) from (25). During the input phase, Node B may select a special vector \( \mathbf{Q} \) that only includes one nonzero element to participate in all calculations from Steps 1 to 4. However, this method does not allow Node B to discriminate \( u + \sum_{i=1}^{n} x_i^2 \) or \( \sum_{i=1}^{n} x_i y_i \) from (25). Unless Node B uses a zero vector, that is, \( \mathbf{Q} = 0 \), it can determine the value of \( u + \sum_{i=1}^{n} x_i^2 \) in Step 3 with considerable probability of success. Using the received value of \( \lambda \), Node B may determine the value of \( r_0 \), after successfully cracking \( u + \sum_{i=1}^{n} x_i^2 \). Even so, Node B cannot find solutions to a group of nonlinear polynomial equations with only \( n + 2 \) equations though the number of unknown quantities is reduced from the original \( 2 n + 3 \) to the current \( 2 n + 2 \).

(4) The security analysis in Step 4 of this process is the same as the analysis in Step 4 of the TPEDP. In order to make the above security analysis more persuadable, it must be proven that the intermediate information obtained by a passive attacker and that produced in no-attack and real situations involve the “calculation of indiscernibility.”

**Theorem 3.** The REDP (marked with \( \pi_3 \)) ensures that Node A and Node B cannot acquire private input regarding each other.

**Proof.** Because of the premises and assumptions of the simulator and because the process of the input phase of the REDP protocol is similar to that of the TPEDP protocol, the validation procedures of computation and output phases may be paid more attention. It can still be supposed that the passive attacker captures Node A and constructs the simulator \( \mathcal{S}_A \) to implement the REDP protocol.

**Computation Phase of \( \pi_3 \).** In Step 1, \( \mathcal{S}_A \) produces \( \{r_0', r_1', \ldots, r_n'\} \), \( u', w' \), and \( e' \) successively according to the results of \( r_1 \) and then calculates \( \{Q_1', Q_2', h_1', h_2', h_3'\} \), which serves as the simulated information that would be sent to Node B.

In Step 2, because Node A does not know the real \( r_0 \) generated by Node B, \( \mathcal{S}_A \) will forge \( r_1' \), together with \( Q' = (y_1', y_2', \ldots, y_n') \) and simulate the intermediate data \( Q_1', Q_2', h_1', h_2', h_3' \), which must be selected by Node B in (13)–(15):
\[
\begin{align*}
Q_1' &= (r_b y_1', r_b y_2', \ldots, r_b y_n'), \\
Q_2' &= (r_b y_1', 2r_b y_2', \ldots, n r_b y_n'), \\
h_1' &= \lambda' r_0' + P_3' \cdot Q_1' = -\frac{1}{2} r_0' r_b' \left( u' + \sum_{i=1}^{n} x_i^2 \right) \\
&\quad + \sum_{i=1}^{n} (r_a' x_i + ir_a' y_i) r_b' y_i', \\
h_2' &= P_3' \cdot Q_2' = \sum_{i=1}^{n} \left( w' - \frac{r_a'}{e'} \right) i r_b' y_i'.
\end{align*}
\]

In Step 3, \( \mathcal{S}_A \) continues to create \( f_a' = u' \) and \( m' = -(h_1' + e' h_2') / r_0' \).

In Step 4, \( \mathcal{S}_A \) obtains \( f_b' = v' = 2 m' / r_0' + \sum_{i=1}^{n} v_i^2 \) in terms of (17) to complete the simulation for Node B.

**Output Phase of \( \pi_3 \).** \( \mathcal{S}_A \) collects all information of Node A: \( \{f_a', f_b', f(\mathbf{P}, \mathbf{Q})\} \), which satisfy the form of (4); that is, \( f_b' = f(\mathbf{P}, \mathbf{Q}) + f_a' \), where \( f_a' = - \varepsilon' \) and \( f_b' = - \varepsilon' \).

\( \mathcal{S}_A \) cannot be distinguished from all information produced in the normal execution of \( \pi_2 \). The view of Node A is \( \text{view}_{\pi_2}(\mathbf{P}, \mathbf{Q}) = \{\mathbf{P}, r_1', h_1, h_2, h_3\} \) and the output of \( \pi_2 \) is \( \text{output}_{\pi_2}(\mathbf{P}, \mathbf{Q}) = d \), so \( \{S_1(\mathbf{P}, d), d\} \equiv \{\text{view}_{\pi_2}(\mathbf{P}, \mathbf{Q}), \text{output}_{\pi_2}(\mathbf{P}, \mathbf{Q})\} \) implies “calculation of indiscernibility.” The same conclusion can be drawn if the participant captured by the attacker is Node B.

This proves Theorem 3.

6. Efficiency Analysis and Comparisons

In this section, the complexity of communication round is analyzed, and the complexity of computation and the type of data involved in various protocols are also evaluated. These protocols obey semihonest execution rules and their objectives are to determine the result of secure distance computation. One of these protocols in previous study [4] is based on homomorphic encryption, called “Rane’s Scheme.” A similar scheme [5], in which Rane’s Scheme was improved
Upon, is called “Rane’s Improved Scheme.” Another previous study [8] proposed two kinds of secure two-party closest-pair computation protocols. The first one comprised the computation of distance aided by a third party, and the distance computation procedure can be separated from the protocol itself and called “Huang’s 1st Scheme.” The second one was designed according to discrete logarithm principles without any third party and could be utilized to realize secure two-party distance computation, called “Huang’s 2nd Scheme.” Lu’s Scheme, Lu’s Scheme, and Li’s Scheme are introduced, respectively, in [6], [7], and [9] in the Related Work section. Lu’s Scheme only realized a scalar product protocol, but, through it, a distance calculation protocol can be simulated at no extra cost. Table 1 lists the results of a comparison of the protocols given above from several perspectives.

(a) Communication Round Complexity. As shown in Table 1, communication round complexities of those protocols are almost roughly the same level except for Li’s Scheme, which includes many communication steps. Rounds of communication in Lu’s Scheme, Rane’s Scheme, and the Rane’s Improved Scheme are determined by the dimensions of the input vector. However, the dimensions of location information must be less than or equal to 3. Even if there is a third party, the TPEDP can be completed within two rounds, but the REDP has a total of four steps and three rounds of interaction, so the complexity of communication round is 3.

(b) Computation Complexity. As shown in Table 1, the overhead of these protocols to generate random numbers can be ignored because that can be addressed in the input phase. The complexity of computation is primarily produced during the computation phase of those protocols, including the TPEDP and REDP. Both of these protocols rely on the dimensions of input vectors, so the computation complexity of them are both \(O(n)\).

Some protocols based on Paillier additive homomorphic encryption can be used to perform \(2 \log N\) rounds of modular multiplication (modular operator \(N^2\)) for each encryption and decryption, at most \(2d\) rounds of modular multiplication (modular operator \(N^2\)) per modular exponential calculation \(E(x)^d\), where \(d\) denotes the number of bits of processed data. For example, the complexity of computation of Rane’s Scheme and its improved one is \(((2n + 2) \log N + 2nd + n + 1)\) rounds of modular multiplication (modular operator \(N^2\)) and that of Lu’s is \(((6n + 2) \log N + (2n + d) + n)\) rounds; it means that the computational cost of Rane’s Scheme and its improved one totally includes \((n + 1)\) times encryptions, \(n\) times modular exponentiations, \(n\) times modular multiplications, and once decryption; Lu’s Scheme is not based on Paillier additive homomorphic encryption but rather on another form of encryption. The complexity of the computation can still be evaluated; it is \((6n \log N + 2nd + 2)\) rounds of modular multiplication (modular operator \(N^2\)). Generally, the value of \(N\) is larger than the dimension \(n\), so the complexity of the computation processes involved in the protocols outlined above is no less than \(O(n^3)\).

(c) Type of Data. Rane’s Scheme shows satisfactory performance with respect to the complexity of rounds of communication, but the types of data that it can process are limited to integer field. Huang’s 2nd Scheme shows the same defect. The two proposed protocols have a broader range of computation because they can be implemented in real number domains.

(d) Degree of Privacy Protection. The TP of Huang’s 1st Scheme holds all the information from both participants, so there is no privacy protection at all. Li’s Scheme, which is based on oblivious transfer protocol and involves no TP, has a primary disadvantage that one participant may acquire all results of distance computation and deduce some private information of the other participant. In this case, if one participant can obtain all three distances from three different starting points to the same terminal point, the exact location of the other participant can be calculated. For example, as shown in Figure 1, if Node A holds \(d_{11}, d_{12},\) and \(d_{13}\), which represent the distances from Node A to Nodes B, C, and D, respectively.
distances from points $T_1$, $T_2$, and $T_3$ to point $H_1$, it can deduce the location of Node B, $H_1$, correctly.

Both the TPEDP and REDP use a result-sharing mechanism in which two participants can determine the distance calculation results until publishing their respective random numbers ($u$ and $v$). This prevents the privacy leakage that can be caused when one participant monopolizes the results. In Section 7.1, especially, it can be proven that the TPEDP protocol's degree of privacy is 1, that is, the perfect zero leakage.

7. Further Discussion

7.1. Privacy Measurement Problem of the TPEDP Protocol. Not all protocols can be carried out and used to measure privacy; only those that always satisfy linear dependence between their inputs and intermediate messages can be evaluated for the degree of privacy by using the methods recommended in Section 3.4. In this way, only the TPEDP protocol meets the requirements outlined in this paper. In the TPEDP protocol, the impact of TP can be completely ignored, because it only satisfies linear dependence between all protocol steps and can be used to measure privacy.

In Section 7.1, especially, it can be proven that the TPEDP protocol's degree of privacy is 1, that is, the perfect zero leakage.

\[
\begin{align*}
T_1 & = \text{Deduction of location.} \\
\end{align*}
\]

\[
s = (P_1 + Q_1) \ast (P_1 + Q_1)^T + 2Rb \ast Q^T - 2Rb \\
\ast P_1^T - q + v \\
= P_1 \ast P_1^T + P_1 \ast Q_1^T + Q_1 \ast P_1^T + Q_1 \ast Q_1^T \\
+ 2Rb \ast Q^T - 2Rb \ast P_1^T - q + v \\
= P_1 \ast P_1^T - 2P_1 \ast Q^T + Rb \ast Rb^T + Q \ast Q^T - q + v.
\]

(27)

The expressions of $p$ and $s$ can be altered as follows:

\[
\begin{align*}
p &= -2Ra \ast Rb^T + Rb \ast Rb^T - q, \\
s &= -2P_1 \ast Q^T + Rb \ast Rb^T + Q \ast Q^T - q + v.
\end{align*}
\]

(28)

These values are related to Node B's inputs, $Q$ and $Rb$. For the convenience of measurement of privacy, the expressions of all three elements can be changed as follows:

\[
\begin{align*}
Q_1 &= I_n \ast Q^T + I_n \ast Rb^T + 0 \ast (Rb \ast Rb^T) \\
& \quad + 0 \ast (Q \ast Q^T) + 0 \ast q + 0 \ast v, \\
p &= 0_{1 \times n} \ast Q^T - 2Ra \ast Rb^T + Rb \ast Rb^T \\
& \quad + 0 \ast (Q \ast Q^T) + 0 \ast q + 0 \ast v, \\
s_1 &= -2P_1 \ast Q^T + 0_{1 \times n} \ast Rb^T + Rb \ast Rb^T \\
& \quad + Q \ast Q^T - q + v.
\end{align*}
\]

(29)

Here, it is supposed that

\[
A_1 = \begin{pmatrix}
I_n & I_n & 0 & 0 & 0 & 0 \\
0_{1 \times n} & -2Ra & 1 & 0 & -1 & 0 \\
-2P_1 & 0_{1 \times n} & 1 & 1 & -1 & 1
\end{pmatrix},
\]

(30)

\[
Z_1 = \begin{pmatrix}
Q^T \\
Rb \\
Q \ast Q^T \\
q \\
v
\end{pmatrix},
C_1 = \begin{pmatrix}
Q^T \\
p \\
s
\end{pmatrix}.
\]

This satisfies $A_1 \ast Z_1 = C_1$ (I$_n$ represents identity matrix). $C_1$ is equivalent to msg$_{r}$ = {$s, p, Q_1$} and rank(A$_1$) is $n + 2$. If any participant obtains $Q$ and msg$_{r}$, he would easily calculate $Rb$, $Rb \ast Rb^T$, $q$, $Q \ast Q^T$, and $v$. According to the definitions given in Section 3.4, it can be concluded that

\[
H(Rb \mid Q, \text{msg}_r) = H(Rb \ast Rb^T \mid Q, \text{msg}_r) = H(q \mid Q, \text{msg}_r) = H(Q \ast Q^T) = H(v \mid Q, \text{msg}_r) = 0 \text{ and } H(Q \mid \text{msg}_r) = H(Q, Rb, Rb \ast Rb^T, Q \ast Q^T, q, v \mid \text{msg}_r).
\]

According to Theorem 1, it can finally be deduced that

\[
H(Q \mid \text{msg}_r) = n \log p.
\]
Next, the degree of privacy is examined from Node B’s perspective. All messages received by Node B can be denoted with \( \text{msg}_{\text{B}} = \{ q, P_1 \} \), where the following are true:

\[
P_1 = I_n \ast P + I_n \ast Ra,
\]

\[
q = (Ra - Rb) \ast (Ra - Rb)^T - p
\]

\[
= Ra \ast Ra^T - 2Rb \ast Ra^T + Rb \ast Rb^T - p.
\]

The expressions of \( q \) and \( P_1 \) can be altered into

\[
P_1 = I_n \ast P + I_n \ast Ra + 0 \ast Ra \ast Ra^T - 0 \ast p,
\]

\[
q = 0_{1 \times n} \ast P - 2Rb \ast Ra^T + Ra \ast Ra^T - p.
\]

Here it is assumed that

\[
A_2 = \begin{pmatrix} I_n & I_n & 0 & 0 \\ 0_{1 \times n} & -2Rb & 1 & -1 \end{pmatrix},
\]

\[
Z_2 = \begin{pmatrix} P \\ Ra^T \\ Ra \ast Ra^T \\ p \end{pmatrix},
\]

\[
C_2 = \begin{pmatrix} P_1 \\ q \end{pmatrix}.
\]

This satisfies \( A_2 \ast Z_2 = C_2 \) (\( I_n \) represents identity matrix). \( \text{Rank}(A_2) \) can be shown to be \( n + 1 \) and \( C_2 \) can be shown to be \( \text{msg}_{\text{B}}, P \) and \( \text{msg}_{\text{B}} \) allow \( Ra, Ra \ast Ra^T, \) and \( p \) to be determined. In this way, it is obvious that \( H(Ra \mid P, \text{msg}_{\text{B}}) = H(\text{Ra} \ast Ra^T \mid P, \text{msg}_{\text{B}}) = H(p \mid P, \text{msg}_{\text{B}}) = 0, H(P \mid \text{msg}_{\text{B}}) = H(P, Ra, Ra \ast Ra^T, p \mid \text{msg}_{\text{B}}), \) and \( H(P \mid \text{msg}_{\text{B}}) = ((2n + 1) - (n + 1)) \log p = n \log p \) in terms of Theorem 1.

Based on the information given above, the following conclusions can be drawn.

**Theorem 4.** The result of privacy measure for the TPEDP is

\[
\text{min}(H(P \mid \text{msg}_{\text{B}}), H(P), H(Q \mid \text{msg}_{\text{B}}), H(Q \mid \text{msg}_{\text{B}}), H(Q)) = 1,
\]

which implies perfect zero leakage.

7.2. The Repeated Calculation Problem of the REDP Protocol.

Repeated calculation problem is worth emphasizing. It merits attention from both of the participants in the REDP protocol. An erroneous choice of participants may lead to a serious reduction in privacy. If Node A and Node B meet again after they have executed the REDP protocol at least once, then, during the new execution, the REDP protocol requires fresh initial vectors and random numbers distinguished from those used in the last calculation. A specific example can be cited and used to illustrate this issue, but discussions are not limited to this example.

Node A and Node B, who held initial vectors \( P \) and \( Q \), respectively, have completed the REDP protocol once. To run the REDP again, Node B here reselects the initial vector \( R \) substituted for \( Q \), but Node A insists on keeping the original vector \( P \) and updates the random sequence \( ra_i \) (\( i = 0, 1, 2, \ldots, n \)) and random numbers \( w \) and \( e \) to construct new \( P_1, P_2, \) and \( \lambda \) values. This indicates that Node B may encounter a group of \( 2n \) nonlinear polynomial equations when it receives those renewed data from Node A. If the new equations in the group are linearly independent from those obtained by Node B in the last execution (Node A held \( P \) and Node B held \( Q \)), then Node B can obtain a group of \( 4n \) nonlinear polynomial equations and \( 2n + 3 + n = 3n + 6 \) unknown quantities. If Node A selects \( ra_0 = 1, e = 1 \) in the new interaction (though this case is almost impossible), the \( 4n \) nonlinear polynomial equations will degenerate into a group of nonhomogeneous linear equations with \( 3n + 6 \) unknown quantities. In this case, if \( n < 6 \) (i.e., \( 4n < 3n + 6 \)), then Node B cannot learn the value of \( P \) because of prohibitively high number of solutions. If \( n \geq 6 \) (i.e., \( 4n \geq 3n + 6 \)) and the number of unknown quantities is less than the number of equations, then Node B will break \( P \) and the location information of Node A is revealed. In this way, if Node A persists in using its original vector, it will be unable to update random sequences and random numbers (such as \( ra, w, \) and \( e \)) in another running of this protocol. Otherwise, Node B will collect more equations regarding this original vector of Node A, which will increase the risk of loss of privacy. However, in this case, Node A can change the random value of \( u \) to produce the corresponding alteration of the results \( f_a \) and \( f_b \).

For this reason, extra limitations must be added to the REDP protocol when it is encoded into the wireless communication devices. This can prevent some extreme situations such as selecting \( ra_0 = 1, e = 1 \) in Node A’s computation round. Certainly, a better provision for the REDP protocol is that each participant should try to choose inputs different from those in previous executions to prevent hidden danger unless the location information of the participant has not changed. (This case has a negligible probability for moving entities.) In this way, Node A must only renew the values of \( u \) or \( v \).

7.3. Availability in Wireless Sensor Networks. It is necessary to verify the availability of TPEDP and REDP in wireless sensor networks. We design such simulation experiments related to distance to observe the changes of network lifetime and energy consumption in the following three scenarios: the first is calculating the distance between Nodes A and B which use the REDP protocol; the second aids TPEDP to achieve the distance calculation (the current cluster-header is “TP”); and the third is based on ECC (elliptic curves cryptography), which is considered as the most lightweight public key method in WSNs, to realize the computation through encryptions, decryptions, and the reliable third party (the base station). To achieve the data-level privacy protection, in the third scenario, Nodes A and B must encrypt their locations separately using the public key of the base station, and the base station decrypts them and calculates the distance between Nodes A and B, and then the base station must encrypt the distance value by their respective public keys and send it to A and B. Finally, Nodes A and B decrypt it and obtain the result. If A and B cannot communicate with the base station within one hop, this method will cause heavy network loads.

In order to ensure the validity of comparisons, we make some assumptions aimed at the three scenarios. Shown in
The second scenario
TP cluster-header

The first scenario
Cluster-header but no TP

Figure 2: Three simulation scenarios.

Figure 2, any two nodes who want to calculate the distance between them in the network can form a group; once a node joins one group it will not be able to join other groups, except the death of the other node in the same group, it will find a new single node with the same situation again to form a new group. In our simulation experiments, nodes would prefer to search the closest neighbor to form a group. All groups repeatedly calculate the distance between two members, until the network life cycle is ended. The base station, especially, does not participate in any group and all packets are in the same size which can be divided exactly by 16. Leach protocol will be used as the clustering and routing protocol in order to ensure path accessibility. If one current cluster-header is dead, it will reselect the new header in terms of Leach protocol. In TPEDP protocol, the same cluster-header can act as the TP for several groups of nodes. Detailed simulation parameters can be shown in Table 2.

We set two cases “n = 3, d = 64 bits” and “n = 8, d = 96 bits” to reflect the trend of network lifetime, based on the comparisons of REDP, TPEDP, and ECC, seen in Figures 3(a) and 3(b), respectively. We can watch that, with the increase of sensor nodes, the network lifetimes of three scenarios (or protocols?) appear to decline (or downtrend?) after a little growth in the initial stage. There exit different reasons of these phenomena for REDP, TPEDP, and ECC. There still exit some active sensor nodes in REDP (or TPEDP) because they cannot find another neighbor (or a suitable TP) in their communication ranges when the number of nodes is 100; however the lifecycle is terminated. With the increase of sensor nodes, the above symptoms will be relieved; for instance, the number of nodes is 150 or 200. With the further raise of the number of nodes (350 or 400), more and more new groups emerge while some previous groups are still running, which causes more communication costs, so the lifetime is shortened. Owing to direct or multihop communication to the base station for each node in the network, the secure distance calculation based on ECC would lead to more energy consumption, compared with REDP and TPEDP. Similarly, 100 nodes are uniformly distributed in 400 * 400 field, however some active nodes have never been able to establish the paths to the base station before their lifecycles end. The suitable enhancement of sensor nodes would create more paths to the base station, which can prolong the network lifetime. However, when the number of nodes is 250 or more, many nodes will play the role of relay to achieve more paths, which causes the decline of lifecycle because of more energy consumption. It is worthy to notice that parameters n and d make have influence on the lifetime of REDP and TPEDP, which in Figure 3(b) is less than Figure 3(a), and the larger the n and d, the more the computation cost of the two protocols. However, the increase of the values of n and d cannot affect the decline of ECC too much, even if it also brings more computational overhead. Compared with TPEDP, REDP causes more energy consumption because of the requirements for more communication rounds and the cost of reselection of new cluster-headers for TP.

7.4. Extended Multiparty Computation. The flexibility characteristics of the TPEDP and REDP protocols make them conveniently transform from two-party distance computation to multi-party distance computation. We suppose that there are totally m nodes in WSNs, and they can be denoted by U1, U2, . . . , Um and hold their secret inputs P1, P2, . . . , Pm, respectively, where P1 = {p11, p12, . . . , p1n}, . . . , Pm = {pm1, pm2, . . . , pmn} are all n-dimensional vectors. In a security multiparty distance protocol, any one of the nodes Ui (i = 1, 2, . . . , m) expects that it can achieve the correct distance
# Simulation parameters setting.

| Parameters                                      | The setting value                                      |
|------------------------------------------------|--------------------------------------------------------|
| Network range                                  | 400 × 400 m²                                           |
| Distribution of sensor nodes                   | Uniform distribution with the number from 100 to 400 nodes, and the base station is in the center |
| Clustering and routing protocol                 | Leach                                                 |
| The number of dimensions of location information| n = 3, n = 8                                           |
| Original energy of node                        | 20 J (infinite energy for the base station)             |
| Energy consumption when sending a packet (at most d = 128 bits) | 0.35 J                                                 |
| Energy consumption when receiving a packet (at most d = 128 bits) | 0.15 J                                                 |
| Energy consumption when processing one bit (at most d = 128 bits) | 0.0005 J                                                |
| Energy consumption when encrypting or decrypting one bit with ECC (at most d = 128 bits) | 0.0005 J                                                |
| Energy consumption when processing a modular multiplication (modular operator N²) (at most d = 128 bits) | 0.001 J                                                 |
| Communication radius of node                   | 50 m                                                   |
| Modular operator N²                             | N = 79, N² = 6241                                      |

**Figure 3:** Comparisons of network lifetime among REDP, TPEDP, and ECC.

The details of calculation are determined by the specific protocol. If \( U_1, U_2, \ldots, U_m \) select their respective random number denoted by \( \theta_1, \theta_2, \ldots, \theta_m \) in the calculation procedure, \( \theta_i = |U_i U_j|^2 + \theta_j \) can be used to represent the final result that keeps the same form of (34). For example, another participant (holds vector \( R \) and random number \( r \)) other than Node A and Node B can easily enjoy the implement of the two proposed protocols, which need to be upgraded simply, and the results would be shared by the form of \( u = |PQ|^2 + v, u = |PR|^2 + r \) and \( v = |QR|^2 + r \).

## 8. Conclusions

Wireless sensor networks make communication and computation between humans and other things become a research hotspot. Two novel secure two-party protocols for distance computation are presented in this paper, designed to address the scenarios of privacy-preservation of location information in an encounter between Node A and Node B. The first proposed protocol, the TPEDP, involves a semihonest TP that protects the location of all participants from a passive attacker inside or outside the network. The problem of nonexistence of a suitable TP in wireless sensor networks was considered. The second protocol lacks a third party. The REDP, which benefits from randomization method, can be used to acquire the correct results of distance between two nodes securely. Compared to the similar protocols, the two proposals, TPEDP...
and REDP, performed more satisfactorily on the efficiency including rounds of communication, computation, and data type.

However, the assumption of the semihonest model is only suitable for certain scenarios, those that can resist only passive attacks. Some protocols secure against the malicious adversary model have been proposed especially for privacy-preserving data applications [20, 25]. However, these techniques are expensive in both communication and computation costs. In the future, the design of efficient protocols that can resist active or malicious attacks may remain challenging.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous reviewers of this paper for their objective comments and helpful suggestions while at the same time helping the authors to improve the English spelling and grammar throughout the manuscript. This research was supported by National Natural Science Foundation of China under Grant nos. 61373138, 61170065, 61202355, 61201163, 61272422, and 61300240, Scientific and Technological Support Project (Industry) of Jiangsu Province under Grant no. BE2012183, Natural Science Key Fund for Colleges and Universities in Jiangsu Province under Grant no. 12KJA520002, Postdoctoral Foundation under Grant no. 2013T60536, and Science and Technology Innovation Fund for Higher Education Institutions of Jiangsu Province under Grant no. CXLX13_467. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the paper.

References

[1] R. H. Weber, “Internet of things—new security and privacy challenges,” Computer Law & Security Review, vol. 26, no. 1, pp. 23–30, 2010.

[2] H. Huang, J. Feng, R. Wang, and X. Qin, “An exact top-k query algorithm with privacy protection in wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 2014, Article ID 749049, 10 pages, 2014.

[3] A. C.-C. Yao, “How to generate and exchange secrets,” in Proceedings of the 27th Annual Symposium on Foundations of Computer Science, pp. 162–167, Toronto, Canada, October 1986.

[4] S. D. Rane, W. Sun, and A. Vetro, “Secure distortion computation among untrusting parties using homomorphic encryption,” in Proceedings of the IEEE International Conference on Image Processing (ICIP ’09), pp. 1485–1488, Cairo, Egypt, November 2009.

[5] S. Rane, W. Sun, and A. Vetro, “Privacy-preserving approximation of L1 distance for multimedia applications,” in Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ’10), pp. 492–497, Singapore, July 2010.

[6] Y. Luo, L. Huang, G. Chen, and H. Shen, “Privacy-preserving distance measurement and its applications,” Chinese Journal of Electronics, vol. 15, no. 2, pp. 237–241, 2006.

[7] R. Lu, X. Lin, and X. Shen, “SPOC: a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency,” IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 3, pp. 614–624, 2013.

[8] H. Huang, H. Zhong, F. Yan, and Y. Sun, “Two protocols for no-information leaked closest-pair of points,” Computer Engineering and Applications, vol. 46, no. 34, pp. 80–81, 2010.

[9] S.-D. Li and Y.-Q. Dai, “Secure two-party computational geometry,” Journal of Computer Science and Technology, vol. 20, no. 2, pp. 258–263, 2005.

[10] W.-G. Zeng, “Efficient 1-out-n oblivious transfer schemes,” in Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC ’02), pp. 159–171, Paris, France, 2002.

[11] M. Rabin, “How to exchange secrets by oblivious transfer,” Tech. Rep. TR-81, Harvard Aiken Computation Laboratory, 1981, http://userpages.umbc.edu/~bhushanl/Day-1.pdf.

[12] M. Ashouri-Talouki, A. Baraani-Dastjerdi, and A. A. Selcuk, “GLP: a cryptographic approach for group location privacy,” Computer Communications, vol. 35, no. 12, pp. 1527–1533, 2012.

[13] Y.-G. Yang, J. Xia, X. Jia, and H. Zhang, “Comment on quantum private comparison protocols with a semi-honest third party,” Quantum Information Processing, vol. 12, no. 2, pp. 877–885, 2013.

[14] M. E. Nergiz, E. Çiçek, T. Pedersen, and Y. Saygin, “A look-ahead approach to secure multiparty protocols,” IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 7, pp. 1170–1185, 2012.

[15] Y. Liu, H. Zhou, and B. Zhao, “Secure location verification using hop-distance relationship in wireless sensor networks,” in Proceedings of the 2nd IEEE Asia-Pacific Service Computing Conference, pp. 62–68, Tsukuba, Japan, December 2007.

[16] M. Xiao, L. Huang, H. Xu, Y. Wang, and Z. Pei, “Privacy preserving hop-distance computation in wireless sensor networks,” Chinese Journal of Electronics, vol. 19, no. 1, pp. 191–194, 2010.

[17] M. Conti, J. Willemsen, and B. Crispo, “Providing source location privacy in wireless sensor networks: a survey,” IEEE Communications Surveys and Tutorials, vol. 15, no. 3, pp. 1238–1280, 2013.

[18] Y. Jian, S. Chen, Z. Zhang, and L. Zhang, “Protecting receiver-location privacy in wireless sensor networks,” in Proceedings of the 26th IEEE International Conference on Computer Communications, pp. 1955–1963, IEEE Computer Society, Anchorage, Alaska, USA, May 2007.

[19] C.-C. Lee, C.-W. Hsu, Y.-M. Lai, and A. Vasilakos, “An enhanced mobile-healthcare emergency system based on extended chaotic maps,” Journal of Medical Systems, vol. 37, no. 5, article 9973, 2013.

[20] Y. Lindell and B. Pinkas, “Secure two-party computation via cut-and-choose oblivious transfer,” Journal of Cryptology, vol. 25, no. 4, pp. 680–722, 2012.

[21] C. Hazay and Y. Lindell, Efficient Secure Two-Party Protocols: Techniques and Constructions, Springer, New York, NY, USA, 2010, http://courses.cs.uu.nl/~lindell/IntroductionEfficient2PC.pdf.

[22] Y. Huang, D. Evans, J. Katz, and L. Malka, “Faster secure two-party computation using garbled circuits,” in Proceedings of the USENIX Security Symposium, p. 35, San Francisco, Calif, USA, 2011, http://www.cs.umd.edu/~jkatz/papers/usenix2011.pdf.
[23] Y.-T. Chiang, D.-W. Wang, C.-J. Liau, and T. Hsu, “Secrecy of two-party secure computation,” in Proceedings of the 19th Annual IFIP WG 1.3 Working Conference on Data and Applications Security, pp. 114–123, Storrs, Conn, USA, 2005.

[24] C.-H. Shen, J. Zhan, D.-W. Wang, T.-S. Hsu, and C.-J. Liau, “Information-theoretically secure number-product protocol,” in Proceedings of the 6th International Conference on Machine Learning and Cybernetics (ICMLC '07), pp. 3006–3011, Corvallis, Ore, USA, August 2007.

[25] J. B. Nielsen, P. S. Nordholt, C. Orlandi, and S. S. Burra, “A new approach to practical active-secure two-party computation,” in Proceedings of the 32nd Annual Cryptology Conference, pp. 681–700, Santa Barbara, Calif, USA, 2012.