International Journal of Engineering

Journal Homepage: www.ije.ir

# Private Trajectory Intersection Testing: Is Garbled Circuit Better than Custom Protocols?

M. Dehghan*[a], B. Sadeghiyan[a], E. Khosravian[b]

[a] *Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran*
[b] *Department of Mechanical Engineering, Payame Noor University, Tehran, Iran*

*PAPER INFO*

*Paper history:*
Received 14 July 2020
Received in revised form 07 January 2021
Accepted 14 January 2021

*Keywords*:
*Distance Computation*
*Euclidean Distance*
*Garbled Circuit*
*Gröbner Basis*
*Private Trajectory Intersection*

*ABSTRACT*

In this paper, two protocols are presented for private intersection detection of two moving objects' trajectories. To design the first protocol, we simplify the problem of finding the intersection points to the problem of finding the common roots of the polynomials, which represent the moving objects' trajectories. Thereafter, Gröbner Basis is used to design a novel secure protocol to find the common roots of the polynomials. Another protocol is also designed based on the distance computation of two trajectories' curves. The complexity of the Gröbner-based protocol for finding the common roots of polynomials is numerical. Its complexity is much lower than the complexity of the garbled circuit-based protocol for Euclidean Distance Computation of $l$ points and the complexity of the protocols for private proximity testing.

***doi:** 10.5829/ije.2021.34.04a.12*

## 1. INTRODUCTION

Private Trajectory Intersection Testing (PTIT) can be considered as a problem in which some moving objects wish to detect the intersection of their trajectories. However, they intend to keep their trajectories secret, namely, to detect whether the distance between moving objects' trajectories is larger than $d$ ($d$ is a predefined threshold), while the parties do not reveal their trajectories of movement.

The problem of PTIT can have some applications in many scenarios, such as urban traffic management, RoboCup competitions, aircraft ad hoc networks, mobile networks and etc. Suppose that the drivers wish to manage traffic in a way to prevent congestion and distribute traffic in streets and highways without revealing their trajectories. Therefore, they want to use a PTIT Protocol. In addition, in RoboCup competitions, the participants tend to select the best trajectories for their robots and do not collide with other robots. These participants apply a PTIT protocol without revealing their trajectories.

The "privacy" is generally in conflict with collecting, storing, using, processing and sharing of personally identifiable data. The primary objective of privacy measures is to ensure proper protection of private data in the course of processing or dissemination of sensitive information [1]. The main feature of PTIT protocol is privacy preserving of moving objects' trajectories.

Secure multiparty computation is an approach to support privacy. In this approach, a set of parties with secret inputs wish to compute some joint functions of their inputs, while they wish to preserve the privacy property. The first general solution for the problem of secure two-party computation in the presence of semi-honest adversaries was presented by Yao [2]. Later, solutions were provided for the multi-party and malicious adversarial cases by Goldreich et al. [3].

In 2016, Hemenway et al. computed the probability of intersection between satellites' trajectories [4]. In their work, the approach is based on garbled circuits, where the garbled circuit for detection of intersection is so large and has not good performance. Atallah and Du

*Corresponding Author Institutional Email: *motahareh479@aut.ac.ir* (M. Dehghan)

[5] presented secure protocols for computational geometry problems. One of the protocols presented in the work of Atallah and Du is for obtaining the intersection of polygons securely. We may use that protocol for obtaining the intersection of moving objects in a certain time, but using it for obtaining the intersection throughout the trajectories is not efficient.

In addition, Frikken and Atallah [6] presented a protocol for two moving objects. In their work, the paths of moving objects are divided into some line segments. Then, in each moment, the intersection of two line segments is obtained. The protocol presented in the work of Frikken and Atallah is just for two-dimensional space.

Moreover, the problem of Private Proximity Testing (PPT) is a similar issue, which is only applicable in smart devices. It has attracted the attention of some researchers since 2006 [7–23]. In PPT, the proximity of two Location Based Services (LBS) are securely examined. Such protocols allow two parties with smart devices, which have Location Based Services to discover if they are close to each other in physical location, without revealing their individual locations to each other. This problem is similar to the problem of PTIT, but in PPT, the parties should have smart devices with LBS.

The previous researches in literature [7–23] are in three categories. In the first category, the parties send securely their location tags to trusted third party, and the trusted third party evaluates if they are neighbor to each other. The second category is based on symmetric key cryptography, and the last is based on public-key infrastructure.

The protocols in the last category are based on modular exponentiation. If we use of RSA for modular exponentiation, we need ‘and’ operations $n^6$ times, ‘or’ operations $n^3 + n$ times. If we want to apply these protocols for PTIT, we should use them for all the points of trajectories, i.e. we need the modular exponentiation $\alpha.n$ times, where $\alpha$ is a constant number.

In this paper, we present two protocols for PTIT of moving objects. In this regard, two protocols are presented based on Gröbner Basis and Distance Computation of two trajectories, which there is no limitation for space dimension. The Gröbner- based protocol was proposed in [24] by Dehghan and Sadeghiyan. We only present an overview of that protocol. Moreover, we analyze its complexity and compare it with the complexity of the garbled circuit-based protocol for Euclidean Distance Computation of *l* points.

The approach proposed by Yao [2] is a ground-breaking result, which essentially began the field of secure multiparty computation and is served as the basis for countless papers. In addition to its fundamental theoretic contribution, Yao’s protocol is remarkably efficient in that it has only a constant number of rounds and uses one oblivious transfer per input bit only (with no additional oblivious transfers in the rest of the computation). However, in the problems with large inputs and circuit size, it might be not efficient.

One way to analyze the complexity of secure multi-party computation protocols, beside the comparison to previous related proposed protocols, is to compare the complexity of the proposed protocols with those of the Yao's garbled circuit-based protocol [2], which is the first general solution.

In this regard, basic operations are implemented using the Yao's garbled circuit [25, 26] (regardless of the proposed security approach) and its computation and communication complexities are examined and compared with the secure proposed protocol.

The basic operation for finding the Intersection of two moving objects’ trajectories regardless security requirement can be considered in such a way that each moving object has *l* discrete points.

Since in Yao's garbled circuit, the circuit gates and their truth tables are formed based on the number of input bits and basic operations, it is necessary to compare the complexity of secure proposed protocols and garbled circuit-based protocols upon the number of input bits and basic operations, as well.

In this paper, we present two protocols for PTIT in Section 2. Then, in Section 3 we prove the security of our proposed protocol based on ideal- real simulation paradigm. We also analyze and compare the computation and communication complexities of our Gröbner-based protocol and garbled cuircuit-based protocol for calculating the Euclidean Distance of *l* points in Section 4.

## 2. THE PROPOSED SECURE PROTOCOLS FOR PRIVATE TRAJECTORY INTERSECTION TESTING

In the proposed secure protocols for two moving objects, the time-dependent polynomial function of the trajectory curve of each moving object should be first computed. Intersection takes place if the distance between two objects at a specific time point, t, is smaller than or equal to d. This constraint is applied to the whole time period and trajectory. Therefore, the trajectory function should also include other points in its proximity (with a radius d) at a specific time, t. Each party considers a point P(x(t), y(t)) on its corresponding curve at time t, preferring that no other parties are present in the proximity (with a radius d) of the point P at a specific time t, as illustrated in Figure 1.

To establish such conditions, one can represent the sur-rounding region (with a radius *d*) of a point at time *t*, as a circle in the two-dimensional space. Time is a key parameter in the proposed protocols, and intersections are considered to occur within the proximity region

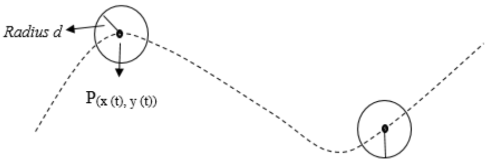

**Figure 1.** The zone of non-intersection of moving objects

(with a radius *d*) of any point of the curve at a specific time, *t*. *P(x(t), y(t))* is the center of circle (as shown in Figure 1) and hence, no other party (i.e. no other point) should be present in this region.

If the function of trajectory be $\vec{f}(t) = x(t).\hat{i} + y(t).\hat{j}$, we express the curve of this trajectory as $(x - x(t))^2 + (y - y(t))^2 = d^2$, which is based on limitation of predetermined threshold *d*.

In the next section, we present two protocols for PTIT, based on Gröbner Basis and Distance Computation.

## 2. 1. The Proposed Secure Protocol I- Based on Gröbner Basis

We want to obtain the intersection of the moving objects' curves, where the curves should be kept private. As the privacy of moving objects' trajectories is important, we generate new curves, where their intersections are the same as the intersections of the moving objects' curves. Moreover, by knowing the new generated curves, no information is obtained from the original curves except their intersections.

Hence, the privacy of the moving objects' trajectories is preserved. Also, in our proposed protocol, the dimension of space is not important.

In this section, we present an overview of a two-party PTIT (PTIT_Gröbner Protocol), which is based on Gröbner Basis. The details of this protocol was presented in Farokhi et al. [13].

***PTIT_ GrÖbner Protocol***

Protocol 1- The PTIT Based on GrÖbner Basis

*party$_1$: selects* $h_1(t)$ *as random, computes* $f_1(t)\times h_1(t)$

1) *Party*$_1$ → *party*$_2$ : $f_1(t)\times h_1(t)$

*party$_2$ : selects* $h_2(t)$ *as random, computes* $f_2(t)\times h_2(t)$

2) *Party*$_2$ → *party*$_1$ : $f_2(t)\times h_2(t)$

*party$_1$: calculates* $V(f_1(t)\times h_1(t)\,,\ f_2(t)\times h_2(t))$, *where* $V(f_1(t)\times h_1(t)\,,\ f_2(t)\times h_2(t)) = ((a_1, a_2, \ldots, a_n), (a'_1, a'_2, \ldots, a'_n), \ldots, (a^{(n)}_1, a^{(n)}_2, \ldots, a^{(n)}_n))$

*calculates* $V(f_1(t)\times h_1(t))$ *and removes them from* $((a_1, a_2, \ldots, a_n), (a'_1, a'_2, \ldots, a'_n), \ldots, (a^{(n)}_1, a^{(n)}_2, \ldots, a^{(n)}_n))$ *where the result is* $V_1$

*party$_2$: calculates* $V(f_1(t)\times h_1(t)\,,\ f_2(t)\times h_2(t))$, *where* $V(f_1(t)\times h_1(t)\,,\ f_2(t)\times h_2(t)) = ((a_1, a_2, \ldots, a_n), (a'_1, a'_2, \ldots, a'_n), \ldots, (a^{(n)}_1, a^{(n)}_2, \ldots, a^{(n)}_n))$

*calculates* $V(,\ f_2(t)\times h_2(t))$ *and removes them from* $((a_1, a_2, \ldots, a_n), (a'_1, a'_2, \ldots, a'_n), \ldots, (a^{(n)}_1, a^{(n)}_2, \ldots, a^{(n)}_n))$ *where the result is* $V_2$

*3) Party*$_1$ $\overset{SSI}{\leftrightarrow}$ *party*$_2$ : *Secure Set Intersection Protocol on* $V_1$ *and* $V_2$

## 2. 2. The Proposed Secure Protocol II- Based on Distance Computation of Curves

In this protocol, the parties first compute securely and jointly the distance function of their trajectories curves as *d(t)*. Each party can check if $d(t) \leq d$, by replacing the corresponding time in distance function *d(t)*. Thus, each party can obtain the collision points by itself.

As it is shown in Protocol 2, the distance function *d(t)* includes three sections. Party$_1$ computes the first section of distance function which is $(x_1(t)^2 + y_1(t)^2)$. Party$_2$ computes the second section of distance function which is $(x_2(t)^2 + y_2(t)^2)$, and the last section should be computed jointly between parties which is $(x_1(t) \times x_2(t) + y_1(t) \times y_2(t))$.

In the first step of protocol, party$_1$ wishes to conceal $(x_1(t)^2 + y_1(t)^2)$. So, it composes an array containing the square of arbitrary values as M, where the $k^{th}$ member of M is $(x_1(t)^2 + y_1(t)^2)$. Also, it adds the values of M by a random array as R. Then, M+R is sent to party$_2$.

In the second step of protocol, party$_1$ hides $(x_1(t), y_1(t))$ using two arrays *A*, *B*, so that the $i^{th}$ member of *A* is $x_1(t)$ and the $i^{th}$ member of *B* is $y_1(t)$. It should be noted that $(x_1(t), y_1(t))$ is used in the last section of distance function. Afterward, party$_1$ sends *A*, *B* to party$_2$.

Party$_2$ computes the multiplication of $(-2 \times x_2(t))$ by *A*, and $(-2 \times y_2(t))$ by *B*. It also adds the results together, and obtains $D = (-2 \times x_2(t) \times A) + (-2 \times y_2(t) \times B)$, that is the last part of distance function. Party$_2$ adds the values of C by a random noise $r_1(t)$.

In the third step of protocol, party$_1$ needs the $i^{th}$ member of array *C*. Therefore, it requests party$_2$ using oblivious transfer. Therefore, party$_2$ sends the $i^{th}$ member of *C*, without disclosure of its index and content.

In the fourth step of the protocol, party$_2$ adds $(x_2(t)^2 + y_2(t)^2) + r_2(t)$ to each member of the array sent by party$_1$ in the first step of protocol and obtains $[(M + R) + (x_2(t)^2 + y_2(t)^2) + r_2(t))\ ;\ 1 \leq i \leq n]$.

As $r_2(t)$ is a random noise function, it can be omitted by filtering. Among these results, party$_1$ just needs the $k^{th}$ member of $[(M + R) + (x_2(t)^2 + y_2(t)^2) + r_2(t))\ ;\ 1 \leq i \leq n]$. So, it requests party$_2$ using oblivious transfer. Party$_2$ sends $(x_1(t)^2 + y_1(t)^2 + r^{(k)}(t)) + (x_2(t)^2 + y_2(t)^2 + r_2(t))$ without disclosure of its index and content.

Finally, party$_1$ computes the sum of $(x_1(t)^2 + y_1(t)^2 + r^{(k)}(t)) + (x_2(t)^2 + y_2(t)^2 + r_2(t))$-$(2 \times x_1(t) \times x_2(t) + 2 \times y_1(t) \times y_2(t) + r_1(t))$, and omits noises $r_1(t)$, $r_2(t)$, $r^{(k)}(t)$ by filtering. Thus, the distance function is computed securely and jointly.

It should be noted that the noises $r_1(t)$, $r_2(t)$ and array *R* are random signals such as white noises, with zero mean and finite variance. They have equal intensity and high fluctuation at different frequencies.

To know more about the Oblivious Transfer Protocol and its extension, an interested reader is referred to [27–30].

***PTIT_Distances Protocol***

Protocol 2- The PTIT Based on Distance Computation

$$d(t)=\sqrt{(x_1(t)-x_2(t))^2+(y_1(t)-y_2(t))^2}=\sqrt{x_1(t)^2+x_2(t)^2-2x_1(t)x_2(t)+y_1(t)^2+y_2(t)^2}$$
$$=\sqrt{-2y_1(t)y_2(t)}=(\underbrace{x_1(t)^2+y_1(t)^2}_{1})+(\underbrace{x_2(t)^2+y_2(t)^2}_{2})\text{-}2\underbrace{(x_1(t)x_2(t)+y_1(t)y_2(t))}_{3}$$

1) $party_1 \rightarrow party_2$ : **M+R** *where* $\mathbf{M} = (a_1(t)^2, a_2(t)^2, \ldots, a_n(t)^2)$;
**R** *is an array of random values, we denote them as* $(r^{(1)}(t), r^{(2)}(t), \ldots, r^{(n)}(t))$
$a_k(t)^2 = x_1(t)^2 + y_1(t)^2$; $1 \le k \le n$

2) $Party_1 \rightarrow party_2$ : $\mathbf{A} = (x'_1(t), x_1''(t), \ldots, x_1^{(n)}(t))$ ;
$x_1^{(i)}(t) = x_1(t)$
$\mathbf{B} = (y'_1(t), y_1''(t), \ldots, y_1^{(n)}(t))$ ; $y_1^{(i)}(t) = y_1(t)$; $1 \le i \le n$
$party_2$: $\mathbf{C} = -2 \times (x_2(t) \times \mathbf{A}) - 2 \times (y_2(t) \times \mathbf{B})$

3) $Party_2 \xleftrightarrow{OT} party_1$ : $-2 \times x_1(t) \times x_2(t) - 2 \times y_1(t) \times y_2(t) + r_1(t)$
$party_2$: $(\mathbf{M}+\mathbf{R}) + (x_2(t)^2 + y_2(t)^2 + r_2(t))$; $r_1(t)$, $r_2(t)$ *are random noises*

4) $party_2 \xleftrightarrow{OT} party_1$ : $(a_k(t)^2 + r^{(k)}(t)) + (x_2(t)^2 + y_2(t)^2 + r_2(t)) = (x_1(t)^2 + y_1(t)^2 + r^{(k)}(t)) + (x_2(t)^2 + y_2(t)^2 + r_2(t))$
*$party_1$: removes noises by filtering and computes* $d^2(t) = (x_1(t)^2 + x_2(t)^2 - 2 \times x_1(t) \times x_2(t) + y_1(t)^2 + y_2(t)^2 - 2 \times y_1(t) \times y_2(t))$

## 3. SECURITY PROOF

In this section, we demonstrate the security proof of Protocol 2, based on the ideal-real simulation paradigm. The security proof of Protocol 1 was demonstrated in in [24].

### 3. 1. Security Proof of PTIT_Distance Protocol

We show that PTIT_Distance Protocol is secure, as long as the employed random generation system and oblivious transfer protocol is secure. Moreover, if the employed random generation system and oblivious transfer protocol are secure against semi-honest adversaries, our PTIT_Distance Protocol is also secure against semi-honest adversaries.

The following Theorem formalizes this statement.

**Theorem 1.** The PTIT_Distance Protocol is fully secure against a semi-honest adversary in the OT-hybrid model.

**Proof.** Our proof follows the ideal/real world paradigm [31]. In particular, we describe a simulator $S_i$ that simulates the ith party's view in the ideal world. Without loss of generality, we describe our security proof for the condition where simulator $S_1$ simulates the view of $party_1$ as Equation (1).

$$(f_1(t), g_1(t), g_2(t)), \tag{1}$$

where $f_1(t)$ is the input of $party_1$ i.e. $(x_1(t), y_1(t))$, and $g_1(t)$ is a random function, which is the output of oblivious transfer protocol in Step 3 instead of $-2 \times x_1(t) \times x_2(t) - 2 \times y_1(t) \times y_2(t) + r_1(t)$. In addition, $g_2(t)$ is a random function, which is the output of oblivious transfer protocol in Step 4 instead of $(x_1(t)^2+y_1(t)^2+r(k)(t))+(x_2(t)^2+ y_2(t)^2+ r_2(t))$.

We now prove Equation (2), which states that the view of party1 can be simulated by a probabilistic polynomial-time algorithm given access to the party1's input and output only. We emphasize that the adversary here is semi-honest, and therefore the view is exactly according to the protocol definition.

$$\{S_1(x, f(x,y))\}_{x,y\in\{0,1\}^*} \overset{c}{\equiv} \{view_1^{\pi}(x,y)\}_{x,y\in\{0,1\}^*} \tag{2}$$

**Indistinguishability of views.** We now prove by contradiction. Assume that there exists a probabilistic polynomial-time distinguisher D and a polynomial p(.) such that, for given *n*,

$$|\Pr[D(H_0(f_1(t), f_2(t), g_1(t), g_2(t)))=1]-\Pr[D(H_n(f_1(t), f_2(t), g_1(t), g_2(t)))=1]|>\frac{1}{p(n)} \tag{3}$$

where $f_1(t)$, $f_2(t)$, $g_1(t)$, $g_2(t)$ are given. $f_1(t)$, $f_2(t)$ are the inputs of $party_1$ and $party_2$, respectively. Also, $g_1(t)$, $g_2(t)$ are instead of the transferred messages.

We now use D to contradict the security of Step 2 of our proposed protocol. First, consider that the only difference between $H_i$ and $H_{i+1}$ is that the (i + 1)th run of this step in $H_i$ is based on $(-2\times x_1(t)\times x_2(t)-2\times y_1(t)\times y_2(t),(x_1(t)^2+y_1(t)^2+r^{(k)}(t))+(x_2(t)^2+y_2(t)^2+r_2(t)))$ and the (i + 1)th run of this step in $H_{i+1}$ is based on $(f_1(t), g_1(t), g_2(t))$, where $g_1(t)$, $g_2(t)$ are random values. Furthermore, given $(f_1(t), f_2(t), g_1(t), g_2(t))$ and a view *v*, it is possible to construct a distribution H such that if *v* is from $(-2\times x_1(t)\times x_2(t)-2\times y_1(t)\times y_2(t),(x_1(t)^2+y_1(t)^2+r^{(k)}(t))+(x_2(t)^2+y_2(t)^2+r_2(t)))$ then H = $H_i$, and if *v* is from $(f_1(t), g_1(t), g_2(t))$, then H = $H_{i+1}$. It therefore follows that it is possible to distinguish the view of $party_1$ in a real execution of our proposed protocol from its simulated view with the same probability that it is possible to distinguish $H_i$ from $H_{i+1}$. However, this contradicts our assumption that $g_1(t)$ and $g_2(t)$ are randomly generated and contradicts the security of the random generation of polynomial functions and oblivious transfer protocol.

## 4. COMPLEXITY ANALYSIS

In this section, we analyze and compare the complexity of PTIT_GrÖbner Protocol and Garbled Circuit-based Protocol for Euclidean Distance Computation of l points.

### 4. 1. Complexity Analysis of PTIT_Grӧbner Protocol

In order to investigate the computational complexity for the PTIT_Grӧbner Protocol, we express the calculations in the protocol based on the binary operations such as 'and', 'or' and 'xor'. In addition, to investigate the communication complexity, we calculate the number of transfer bits in each step of PTIT_ Gröbner Protocol. In the following, further details are expressed to analyze the complexity of the proposed protocol for PTIT_Gröbner Protocol. The steps outlined below are based on Protocol 1.

**Step 1**- Party$_1$ sends $f_1(t) \times h_1(t)$ to party$_2$. To transfer these arrays, it is necessary to send the coefficients of these polynomials in an L member array. Here, L is a security parameter and both parties are aware of its size. Since it is possible to have floating point coefficients in polynomials, each party can display the coefficients with a good accuracy using 32 bits. We consider the number of bits of the coefficients is equal to *m*. Thus, the transfer array is a (m × L)-bit array. It is also necessary to multiply two polynomials, which is multiplication of two (m × L)-bit arrays. This multiplication requires performing m × m × L × L times 'and' operations and m × L times 'or' operations, and should be done for each member of the array.

**Step 2 -** Similarly, party$_2$ sends $f_2(t) \times h_2(t)$ to party$_1$. Similarly, the transfer array is a (m × L)-bit array. It is also necessary to multiply two polynomials, which is multiplication of two (m×L)-bit arrays. This multiplication requires performing m × m × L × L 'and' operations and m × L 'or' operations.

**Step 3** - Then, each party calculates the varieties of $h_2(t)\times f_2(t)$ and $h_1(t)\times f_1(t)$. The varieties are calculated using the Buchberger's algorithm [32], which consists of a number of polynomial divisions. The polynomial division consists of binary multiplication and addition, which is reduced to modulo *n* at each step. Furthermore, since the coefficients of the polynomials have *m* bits, the division operation is performed in G(m$^n$).

In each division operation, it is necessary to perform the basic 'and / or' operations for a number of times (which is equal to the bits of coefficients). Since the coefficients are *m* bits, it is necessary to perform 'and' , 'or' operations, *m*$^2$ and *m* times, respectively. In this manner, the Gröbner Basis of Ideal $< f_1(t) \times h_1(t), f_2(t), h_2(t) >$ is calculated. To find all bases, it is necessary to perform the division operation multiple times, equal to the number of ideal bases, which is assumed to be *r*. In other words, we should perform 'and/ or' operations L × (m$^2$ + m) × r times. This is carried out by each of the parties.

**Step 4** - The first party eliminates the roots of h$_1$(t) from the varieties of $< f_1(t) \times h_1(t), f_2(t), h_2(t) >$, and keeps the result secret. In fact, the first party acquires the common roots of $f_1(t)$ and $f_2(t) \times h_2(t)$. In this step and the fifth step, if we consider the number of ideal bases to be equal to s, L × (m$^2$ + m) × s basic 'and' or 'or' operations should be performed. The third, fourth, and fifth steps are performed without communication complexity, being only performed with computation complexity.

**Step 5** - The second party removes the roots of h$_2$(t) from the varieties of $< f_1(t) \times h_1(t), f_2(t), h_2(t) >$, and keeps the result secret, as well. Indeed, the second party obtains the common roots of $f_2(t)$ and $f_1(t) \times h_1(t)$. As stated in the fourth step, in this step, L × (m$^2$ + m) × s basic 'and' or 'or' operations should be performed.

**Step 6** - Finally, both parties participate in the secure set intersection protocol, and securely obtain the common roots of f$_1$(t) and f$_2$(t). This step is done in combination, using a trusted third party. To implement this hybrid protocol, each party maintains its status in the main protocol as $\sigma_i$, and then executes the $\rho_i$ protocol that securely calculates the intersection of sets. After receiving the output of $\rho_i$, they return to $\sigma_i$. To express the computation complexity of $\rho_i$, we use the secure protocol proposed in the literature [33]. In that protocol, each party forms a polynomial from the members of its own set, so that the members of the set are the roots of polynomial $(f_1(x) = (x-\alpha_1)(x-\alpha_2)...(x-\alpha_{k_1}))$. If the number of the set members is equal to k1, then the degree of f$_1$(x) is also equal to k$_1$. Then, each party selects a random polynomial r(x). Each party shares the coefficients of the polynomial h(x)=f(x).r(x) to the other party. Each party can use the share values to create a part of the polynomial u(x)=h$_1$(x) + h$_2$(x) and send it to the other party. With calculating the final polynomials, the parties can substitute the members of their own sets into the polynomial. If the evaluation is zero then the element belongs to the intersection, else the element does not belong to the intersection. These calculations are a number of addition and multiplication operations that can be converted to 'and /or' operations. The numbers of addition and multiplication operations is in order of O(k$_1$ + k$_2$), where k$_1$ is the number of set members of the first party, and k$_2$ is the number of set members and the polynomial degree of the second party, respectively. If we represent each coefficient with m bits, each addition operation can be converted to m times 'or' operations. Moreover, each multiplication operation can be converted to m times 'or' operations and m$^2$ times 'and' operations. Finally, the computation complexity of the PTIT_Gröbner Protocol is in order of O(m$^2$ × (k$_1$ + k$_2$)). In the worst case, the size of the two sets are equal, which can be considered to be k. In this case, the computation complexity is in order of O(m$^2$ × k). To calculate the number of transfer bits between parties, we need to calculate the number of bits to share the coefficients of polynomials. Each party creates the coefficient shares and then, sends it to the other party. This action is done twice by each party and hence, a total of 2 × (k$_1$ + k$_2$) × m bits are sent. In general, the communication complexity of secure set intersection protocol is in order of O(k × m). Furthermore, each party enters its own set into $\rho_i$ protocol such that the size of inputs is 2 × k × m bits at worst.

Finally, the computation complexity of PTIT_Gröbner Protocol is as follows, which indicates the number of 'and , or' operations: $2 \times (m^2 + m) \times L + 2 \times (m^2 + m) \times r \times L + 2 \times (m^2 + m) \times s \times L + m^2 \times k \approx O(m^2 \times L \times (r + s + k)) \approx O(m^2 \times L^2 \times Q)$.

In the above computation complexity, L indicates the array size of the polynomial coefficients, r and s denote the number of the bases generating the ideals of polynomials in the third to fifth steps, and k represents the number of varieties that enter the secure set intersection protocol. We denote the summation of these variables with Q.

If the degree of polynomials is finite, the number of polynomial coefficients, which is denoted as L, has also a finite size. On the other hand, since the degree of ideals and the number of variables are finite, the number of bases of ideals in the third to fifth steps is finite, too. In addition, the number of input bits for displaying the location coordinates is finite as well, such that decimal numbers can be displayed with a good accuracy using 32 bits. If the degree of polynomials are assumed to be finite, the computation complexity of this protocol is of numerical order.

The communication complexity of the protocol steps is as 2 × m × L + 3 × k × m ≈ O((k + L) × m) transfer bits, in which L is the size of the coefficients array. Also, k is the number of varieties, which are the inputs of the secure set intersection protocol, and m is the number of the coefficients bits, which are used in the secure set intersection protocol. If we assume that the degree of the polynomials is finite, L and k are finite. Moreover, if we consider the number of bits used for displaying the coefficients to be 32 bits, the communication complexity of this protocol is numerical.

### 4. 2. Complexity Analysis of Garbled Circuit-based Protocol for Euclidean Distance Computation of l Points

One way to analyze the complexity of secure multi-party computation protocols is to compare the complexity of the proposed protocols with those of the Yao's garbled circuit-based protocols [2].

The basic operation for finding the intersection of two moving objects' trajectories regardless of security requirement can be considered in such a way that each moving object has l discrete points.

In this regard, basic operations are implemented using the Yao's garbled circuit [25, 26] (regardless of the proposed security approach) and its computation and communication complexities are examined and compared with the secure proposed protocol.

The details of Yao's garbled circuit have been previously presented [25, 26]. In this section, we express the computation and communication complexities of this circuit to calculate the Euclidean Distance of *l* points.

To create a garbled circuit for calculating the Euclidean Distance, it is necessary to consider the number of input bits and the number of base operations for computing the Euclidean Distance. The number of bits for representation of the points P and Q are considered to be equal to m. Since x and y coordinates are used to represent spatial points (P and Q), it is necessary to display each coordinate with m bits and calculate the Euclidean distance as Equation (4).

$$d(t)=\sqrt{(x_1(t)-x_2(t))^2+(y_1(t)-y_2(t))^2} = \{x_1(t)^2+x_2(t)^2+y_1(t)^2+y_2(t)^2 - 2\times(x_1(t)x_2(t)+y_1(t)y_2(t))\}^{\frac{1}{2}} \quad (4)$$

Each party performs the square operation on its side. Then, (m+1)-bit addition carried out five times, and m-bit multiplication are done two times, where m is the number of input bits. The (m + 1)-bit addition includes m + 1 times 'or' operations, and the m-bit multiplication includes $m^2$ times 'and', and also m times 'or' operations.

The sender creates a circuit with a number of gates described. It then creates a truth table for each gate and garbles it. For garbling, it is necessary to create independent keys for the inputs and outputs of 'and' operations ($2m^2$ times) and 'or' operations (5m + 5 times). In fact, it creates $6 \times (5m + 5 + 2m^2) \approx 12m^2$ keys, which are n-bit keys. Since these gates include 'and / or' gates, it is necessary to create keys and garbled truth tables for all gates.

For each gate, the sender creates a garbled truth table that includes output keys encrypted using the corresponding input keys. According to the Half Gates method [34] that uses Row Reduction [35], for each gate, it is necessary to perform the modular exponentiation only twice, and send two n-bit messages that are the outputs of the modular exponentiation. Indeed, $2 \times (5m + 5 + 2m^2) \times (n^6 + n^3 + n)$ 'and' and 'or' operations are required to create a garbled truth table. Furthermore, $2 \times (5m + 5 + 2m^2) \times n$ bits are transferred.

For each gate, the sender sends one n-bit key corresponding to its input to the receiver. The number of these gates is $(5m + 5 + 2m^2)$ and for each gate, one n-bit key is transferred. The number of transfer bits equals to $(5m + 5 + 2m^2) \times n$.

In addition, in the oblivious transfer protocol, the sender and receiver transfer a finite number of bits, and the final output of the oblivious transfer protocol is an n-bit key. For each 'and / or' gate, an oblivious transfer protocol is performed, and one n-bit key is transferred. Therefore, a total of $(5m + 5 + 2m^2) \times n$ bits are transferred. Thus, the communication complexity of the protocol for finding the Euclidean Distance between two m-bit points using garbled circuit equals: 4 × n ×

$(5m + 5 + 2m^2) = 20 \times n \times m + 20 \times n + 8 \times m^2 \times n \approx O(m^2 \times n)$ bits.

To calculate the computation complexity, we consider the calculations that the sender and receiver do on their side. The sender generates the C circuit, which describes the Euclidean Distance function. It includes $6m + 4$ 'or' gates and $2m^2$ 'and' gates. It then creates n-bit keys for each input and keeps them secret. As previously described, a modular exponentiation algorithm such as RSA is used to create the n-bit keys. So, it is required to perform 'and' operations $n^6$ times, and 'or' operations $n^3 + n$ times.

The sender generates six keys for each 'and / or' gates, where four keys are generated for inputs and two keys are for output. Totally, the sender generates $6 \times (6m + 4 + 2m^2)$ n-bit keys. Creating such keys requires $6 \times (6m + 4 + 2m^2) \times (n^3 + n + n^6) \approx O(n^6 \times m^2)$ 'and / or' operations.

For each gate, the sender creates a garbled truth table that includes output keys encrypted using the corresponding input keys. He uses a modular exponentiation algorithm such as RSA to encrypt the output keys. We assume that the input and output keys and all security parameters have n bits. The sender can use the fast modular exponentiation algorithm for encrypting, which requires $n^6$ 'and' operations and $n^3 + n$ 'or' operations for creating each member of the garbled truth table. The number of key bits generated by the sender equals $O(n^6 \times m^2) \approx (n^6 + n^3 + n) \times 6 \times (6m + 2m^2)$ bits.

In each implementation of the oblivious transfer protocol, the sender and receiver perform three 'xor' operations on their own sides. We apply Free Xor [36, 37] method to reduce the communication cost per each 'xor' gate. In addition, in Free Xor method the number of generated keys per 'xor' gate, the number of transferred keys, and the number of encryption and decryption are zero. In fact, the garbled 'xor' operation is converted to a simple 'xor'.

In addition, the permutation is executed three times. If we consider the permutation as the modular exponentiation, and use the fast modular exponentiation algorithm, 'and' operation should be done $n^6$ times, and 'or' operations should be done $n^3 + n$ times. Thus, for each gate, an oblivious transfer protocol is performed, where $(n^6 + n^3 + n) \times (6m + 2m^2) \approx O(n^6 \times m^2)$ times 'and / or' operations are done for it.

In addition, the sender and receiver perform the x and y square operations, including multiplication and addition operations. If this operation is performed in a binary manner, for each m-bit multiplication operation, $m^2$ times 'and', m times 'or' operations are performed. On the other hand, for each m-bit addition operation, m times 'or' operations are performed. Therefore, a total of $2 \times (m^2 + 2m) \approx m^2$ 'and/or' operations would be generally needed.

Finally, the computation complexity of Yao's garbled circuit for calculating the Euclidean Distance of two points is in order of $n^6 \times m^2$. The computation and communication complexity that we have expressed is for calculating the Euclidean Distance of two m-bit points. On the other hand, the trajectory of moving objects includes *l* points, which are m-bit. It is necessary to perform the above operation *l* times. Therefore, the computation complexity is in order of $n^6 \times m^2 \times l$, and the communication complexity is in order of $m^2 \times n \times l$, depending on the number of input bits, key size and the number of trajectory points. Increasing these three parameters will increase the computation and communication complexities.

### 4. 3. Complexity Comparison of PTIT_Gröbner Protocol and Garbled Circuit_based Protocol

The input bits represent the x and y coordinates of the trajectory points. These points can be represented by decimal numbers, which can be accurately indicated with 32 bits.

We can consider all variables (except for the security parameter, n) to be equal, due to the limitation and small growth compared to n, and calculate the complexities. In this manner, the computation complexity of the garbled circuit- based protocol is in order of $n^6$, and the order of its communication complexity is n. In contrast, the computation and communication complexities of the PTIT_Gröbner protocol is numeric.

As illustrated in Table 1, the PTIT_ Gröbner protocol is better than the garbled circuit-based protocol. Moreover, the PTIT_Distance Protocol is based on oblivious transfer protocol. Therefore, it is based on modular exponentiation, and we need 'and' operations $n^6$ times, and 'or' operations $n^3 + n$ times. Thus, due to high growth of n, its complexity is almost the same as the complexity of garbled circuit-based protocol.

As stated before, the problem of PPT is similar to PTIT. However, in PPT the parties should have smart devices with LBS. Thus, to compare the complexity of our proposed protocols and the previous decentralized protocols for PPT [11, 18–22], we demonstrate their complexities in Table 2.

All the mentioned protocols in Table 2 are based on modular exponentiation, and they need 'and' operations $n^6$ times, and 'or' operations $n^3 + n$ times, for each modular exponentiation. A comparison between Table 1 and Table 2 shows that PTIT_Gröbner protocol is better than the garbled circuit-based protocol for Euclidean Distance Computation of *l* points and the previous researches for PPT.

**TABLE 1.** Complexity comparison of our proposed protocols and garbled circuit-based protocol for Euclidean Distance calculation

| Communication Complexity | Computation Complexity | Protocol |
|---|---|---|
| O($m^2$×n×l) | O($n^6$×$m^2$×l) | Calculation of the Euclidean Distance of *l* points based on the Yao's garbled circuit |
| O(m×(k+L)) | O($m^2$ × L ×Q) | PTIT Protocol based on GrÖbner Basis (PTIT_ GrÖbner Protocol) |
| O(N×m) | O($n^6$ ×m× N) | PTIT Protocol based on Distance Computation (PTIT_Distance Protocol) |

n: The number of bits for input-output keys and security parameters

m: The number of input bits

Q: The sum of s + r + k, related to three variables, representing the number of varieties and the number of ideals at different stages.

k: The number of input varieties of the secure set intersection protocol (also, the degree of polynomials in the secure set intersection protocol)

L: The size of the polynomial coefficients array

N: The size of transferred array for concealing the trajectory points

**TABLE 2.** Complexity comparison of the proposed protocols for PPT [11, 13-15, 17, 22]

| Number of Base Operations | Number of Exponentiation Operations | Cryptography Algorithm | Protocol |
|---|---|---|---|
| For each exponentiation, $n^6$ '*and*' operation, $n^3$+n '*or*' operation is required.<br>Alice: 6×( $n^6$ +$n^3$+n)+ 3 DL<br>Bob: 6×( $n^6$ +$n^3$+n) | Alice: 6 exp+3 DL<br>Bob: 6 exp | CGS | Pierre, 2007 [18] |
| 3×( $n^6$ +$n^3$+n) | 3 exp/user | DH-based SPEKE | NFP, 2009 [20] |
| Alice: 3×( $n^6$ + $n^3$+n)<br>Bob: 4×( $n^6$ +$n^3$+n) | Alice: 3 exp<br>Bob: 4 exp | ElGamal encryption | EG-PET , 2011 [19] |
| 3×( $n^6$ +$n^3$+n) | 3 exp/user | DH-based SPEKE | DH-PET, 2014 [21] |
| 2×($n^6$ +$n^3$+n) | 2 exp/user | RSA | FP-PET , 2016 [22] |
| Alice: 3×( $n^6$ + $n^3$+n)<br>Bob: 2×( $n^6$ +$n^3$+n) | Alice: 3 exp<br>Bob: 2 exp | DGK | FPPLP, 2018 [11] |

## 5. CONCLUSION

In this paper, we presented two protocols for private intersection detection of two moving objects' trajectories, which are based on Gröbner Basis and Distance Computation. The Gröbner-based protocol was proposed in a previous research, and we only presented its overview. We also presented the complexity analysis of the Gröbner-based protocol. We compared its complexity by the garbled circuit-based protocol for Euclidean Distance Computation of *l* points. This approach for complexity analysis is common in the field of security protocol design. Moreover, we proved the security of the proposed protocol, which is based on Distance Computation.

Our comparison demonstrated that PTIT_Gröbner protocol is better than the garbled circuit-based protocol for Euclidean Distance Computation of l points and the previous researches for PPT.

## 6. REFERENCES

1. Asadi Saeed Abad, F., and Hamidi, H., "An Architecture for Security and Protection of Big Data", ***International Journal of Engineering - Transaction A: Basics***, Vol. 30, No. 10, (2017), 1479–1486. doi:10.5829/ije.2017.30.10a.08
2. Yao, A. C., "Protocols for secure computations", 23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982), (1982), IEEE, 160–164. doi:10.1109/SFCS.1982.38
3. Goldreich, O., Micali, S., and Wigderson, A., "How to play ANY mental game", Proceedings of the Nineteenth Annual ACM Conference on Theory of Computing - STOC '87, (1987), 218–229. doi:10.1145/28395.28420
4. Hemenway, B., Lu, S., Ostrovsky, R., and Welser IV, W., "High-Precision Secure Computation of Satellite Collision Probabilities", In International Conference on Security and Cryptography for Networks, Springer, Cham, (2016), 169–187. doi:10.1007/978-3-319-44618-9_9
5. Atallah, M. J., and Du, W., "Secure Multi-party Computational Geometry", In Workshop on Algorithms and Data Structures, Springer, Berlin, Heidelberg, (2001), 165–179. doi:10.1007/3-540-44634-6_16

6. Frikken, K. B., and Atallah, M. J., "Privacy preserving route planning", Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society - WPES '04, (2004). doi:10.1145/1029179.1029182

7. Cheng, R., Zhang, Y., Bertino, E., and Prabhakar, S., "Preserving User Location Privacy in Mobile Data Management Infrastructures", In International Workshop on Privacy Enhancing Technologies, Springer, Berlin, Heidelberg, (2006), 393–412. doi:10.1007/11957454_23

8. Mascetti, S., Freni, D., Bettini, C., Wang, X. S., and Jajodia, S., "Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies", ***The VLDB Journal***, Vol. 20, No. 4, (2011), 541–566. doi:10.1007/s00778-010-0213-7

9. Zhu, H., Wang, F., Lu, R., Liu, F., Fu, G., and Li, H., "Efficient and Privacy-Preserving Proximity Detection Schemes for Social Applications", ***IEEE Internet of Things Journal***, Vol. 5, No. 4, (2018), 2947–2957. doi:10.1109/JIOT.2017.2766701

10. Zheng, Y., Li, M., Lou, W., and Hou, Y. T., "Location Based Handshake and Private Proximity Test with Location Tags", ***IEEE Transactions on Dependable and Secure Computing***, Vol. 14, No. 4, (2017), 406–419. doi:10.1109/TDSC.2015.2472529

11. Järvinen, K., Kiss, Á., Schneider, T., Tkachenko, O., and Yang, Z., "Faster Privacy-Preserving Location Proximity Schemes", In International Conference on Cryptology and Network Security,Springer, Cham, (2018), 3–22. doi:10.1007/978-3-030-00434-7_1

12. Pagnin, E., Gunnarsson, G., Talebi, P., Orlandi, C., and Sabelfeld, A., "TOPPool: Time-aware Optimized Privacy-Preserving Ridesharing", ***Proceedings on Privacy Enhancing Technologies***, Vol. 2019, No. 4, (2019), 93–111. doi:10.2478/popets-2019-0060

13. Farokhi, F., Shames, I., and Johansson, K. H., "Private routing and ride-sharing using homomorphic encryption", ***IET Cyber-Physical Systems: Theory & Applications***, Vol. 5, No. 4, (2020), 311–320. doi:10.1049/iet-cps.2019.0042

14. Oleynikov, I., Pagnin, E., and Sabelfeld, A., "Where Are You Bob? Privacy-Preserving Proximity Testing with a Napping Party", In European Symposium on Research in Computer Security, Springer, Cham, (2020), 677–697. doi:10.1007/978-3-030-58951-6_33

15. Chor, B., Kushilevitz, E., Goldreich, O., and Sudan, M., "Private information retrieval", ***Journal of the ACM***, Vol. 45, No. 6, (1998), 965–981. doi:10.1145/293347.293350

16. Siksnys, L., Thomsen, J. R., Šaltenis, S., and Yiu, M. L., "Private and Flexible Proximity Detection in Mobile Social Networks", 2010 Eleventh International Conference on Mobile Data Management, (2010), IEEE, 75–84. doi:10.1109/MDM.2010.43

17. Šikšnys, L., Thomsen, J. R., Šaltenis, S., Yiu, M. L., and Andersen, O., "A Location Privacy Aware Friend Locator", In International Symposium on Spatial and Temporal Databases, Springer, Berlin, Heidelberg., (2009), 405–410. doi:10.1007/978-3-642-02982-0_29

18. Zhong, G., Goldberg, I., and Hengartner, U., "Louis, Lester and Pierre: Three Protocols for Location Privacy", In International Workshop on Privacy Enhancing Technologies, Springer, Berlin, Heidelberg, (2007), 62–76. doi:10.1007/978-3-540-75551-7_5

19. Narayanan, A., Thiagarajan, N., Lakhani, M., Hamburg, M., and Boneh, D., "Location Privacy via Private Proximity Testing", In NDSS, Vol. 11, (2011).

20. Chatterjee, S., Karabina, K., and Menezes, A., "A New Protocol for the Nearby Friend Problem", In IMA International Conference on Cryptography and Coding, Springer, Berlin, Heidelberg, (2009), 236–251. doi:10.1007/978-3-642-10868-6_14

21. Magkos, E., Kotzanikolaou, P., Magioladitis, M., Sioutas, S., and Verykios, V. S., "Towards Secure and Practical Location Privacy through Private Equality Testing", In International Conference on Privacy in Statistical Databases, Springer, Cham, (2014), 312–325. doi:10.1007/978-3-319-11257-2_24

22. Kotzanikolaou, P., Patsakis, C., Magkos, E., and Korakakis, M., "Lightweight private proximity testing for geospatial social networks", ***Computer Communications***, Vol. 73, (2016), 263–270. doi:10.1016/j.comcom.2015.07.017

23. Šeděnka, J., and Gasti, P., "Privacy-preserving distance computation and proximity testing on earth, done right", Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, (2014), 99–110. doi:10.1145/2590296.2590307

24. Dehghan, M., and Sadeghiyan, B., "Privacy-preserving collision detection of moving objects", ***Transactions on Emerging Telecommunications Technologies***, Vol. 30, No. 3, (2019), e3484. doi:10.1002/ett.3484

25. Huang, Y., Evans, D., and Katz, J., "Private set intersection: Are garbled circuits better than custom protocols?", In 19th Network and Distributed Security Symposium, San Diego, (2012), 1–15.

26. Bellare, M., Hoang, V. T., and Rogaway, P., "Foundations of garbled circuits", Proceedings of the 2012 ACM Conference on Computer and Communications Security - CCS '12, (2012), 784–796. doi:10.1145/2382196.2382279

27. Rabin, M. O., "How To Exchange Secrets with Oblivious Transfer", IACR Cryptol. EPrint Arch., (2005).

28. Yao, A. C.-C., "How to generate and exchange secrets", 27th Annual Symposium on Foundations of Computer Science (Sfcs 1986), (1986), IEEE, 162–167. doi:10.1109/SFCS.1986.25

29. Naor, M., and Pinkas, B., "Efficient oblivious transfer protocols", In Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Washington DC (Vol. 1), (2001), 448–457. doi:10.1145/365411.365502

30. Ishai, Y., Kilian, J., Nissim, K., and Petrank, E., "Extending Oblivious Transfers Efficiently", In Annual International Cryptology Conference, Springer, Berlin, Heidelberg, (2003), 145–161. doi:10.1007/978-3-540-45146-4_9

31. Lindell, Y., and Pinkas, B., "A Proof of Security of Yao's Protocol for Two-Party Computation", ***Journal of Cryptology***, Vol. 22, No. 2, (2009), 161–188. doi:10.1007/s00145-008-9036-8

32. Cox, D. A., Little, J., and O'Shea, D., Ideals, Varieties, and Algorithms, Cham, Springer International Publishing, (2015). doi:10.1007/978-3-319-16721-3

33. Li, R., and Wu, C., "An Unconditionally Secure Protocol for Multi-Party Set Intersection", In International Conference on Applied Cryptography and Network Security, Springer, Berlin, Heidelberg, (2007), 226–236. doi:10.1007/978-3-540-72738-5_15

34. Zahur, S., Rosulek, M., and Evans, D., "Two Halves Make a Whole", In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, (2015), 220–250. doi:10.1007/978-3-662-46803-6_8

35. Pinkas, B., Schneider, T., Smart, N. P., and Williams, S. C., "Secure Two-Party Computation Is Practical", In International Conference on the Theory and Application of Cryptology and Information Security, Springer, Berlin, Heidelberg, (2009), 250–267. doi:10.1007/978-3-642-10366-7_15

36. Kolesnikov, V., and Schneider, T., "Improved Garbled Circuit: Free XOR Gates and Applications", In International Colloquium on Automata, Languages, and Programming, Springer, Berlin, Heidelberg., (2008), 486–498 Berlin, Heidelberg, Springer Berlin Heidelberg. doi:10.1007/978-3-540-70583-3_40

37. Kolesnikov, V., Mohassel, P., and Rosulek, M., "FleXOR: Flexible Garbling for XOR Gates That Beats Free-XOR", In Annual Cryptology Conference, Springer, Berlin, Heidelberg, (2014), 440–457. doi:10.1007/978-3-662-44381-1_25

## Persian Abstract

*چکیده*

ما در این مقاله دو پروتکل برای یافتن امن اشتراک مسیر حرکت دو شیء متحرک ارائه می‌دهیم. برای طراحی پروتکل اول، مسئله یافتن امن اشتراک مسیر حرکت را به مسئله یافتن امن ریشه مشترک دو چند جمله‌ای نمایش دهنده مسیر حرکت تبدیل می‌نماییم. سپس، از پایه گروبنر برای طراحی پروتکل یافتن امن ریشه مشترک دو چند جمله ای استفاده می‌کنیم. همچنین، پروتکل دیگری بر مبنای محاسبه فاصله دو منحنی مسیر حرکت ارائه می‌دهیم. پروتکل یافتن امن ریشه مشترک دو چند جمله‌ای بر مبنای پایه گروبنر، دارای پیچیدگی عددی است و در مقایسه با پروتکل مبتنی بر مدار مبهم برای محاسبه فاصله اقلیدسی $l$ نقطه و پروتکل های پیشین برای بررسی امن نزدیک بودن اشیاء متحرک، دارای پیچیدگی بسیار پایین‌تری است.