Secure Multi-Party Computation Problems and Their Applications: A Review and Open Problems [*†]

Wenliang Du
Center for Systems Assurance
Department of Electrical Engineering and Computer Science
Syracuse University
121 Link Hall
Syracuse, NY 13244
wedu@ecs.syr.edu

Mikhail J. Atallah
Department of Computer Sciences and Center for Education and Research in Information Assurance and Security
Purdue University
1315 Recitation Building
West Lafayette, IN 47907
mja@cs.purdue.edu

ABSTRACT

The growth of the Internet has triggered tremendous opportunities for cooperative computation, where people are jointly conducting computation tasks based on the private inputs they each supplies. These computations could occur between mutually untrusted parties, or even between competitors. For example, customers might send to a remote database queries that contain private information; two competing financial organizations might jointly invest in a project that must satisfy both organizations' private and valuable constraints, and so on. Today, to conduct such computations, one entity must usually know the inputs from all the participants; however if nobody can be trusted enough to know all the inputs, privacy will become a primary concern.

This problem is referred to as Secure Multi-party Computation Problem (SMC) in the literature. Research in the SMC area has been focusing on only a limited set of specific SMC problems, while privacy concerned cooperative computations call for SMC studies in a variety of computation domains. Before we can study the problems, we need to identify and define the specific SMC problems for those computation domains. We have developed a framework to facilitate this problem-discovery task. Based on our framework, we have identified and defined a number of new SMC problems for a spectrum of computation domains. Those problems include privacy-preserving database query, privacy-preserving scientific computations, privacy-preserving intrusion detection, privacy-preserving statistical analysis, privacy-preserving geometric computations, and privacy-preserving data mining.

The goal of this paper is not only to present our results, but also to serve as a guideline so other people can identify useful SMC problems in their own computation domains.

Keywords

Privacy, secure multi-party computation.

1. INTRODUCTION

The proliferation of the Internet has triggered tremendous opportunities for cooperative computation, where people are cooperating with each other to conduct computation tasks based on the inputs they each supplies. These computations could occur between trusted partners, between partially trusted partners, or even between competitors. For example, customers might send to a remote database the queries that contain private information, two competing financial organizations might jointly invest in a project that must satisfy both organizations' private and valuable constraints, and so on. Usually, to conduct these computations, one must know inputs from all the participants; however if nobody can be trusted enough to know all the inputs, privacy will become a primary concern. For example, consider the following applications:

1. Alice thinks that she may have some genetic disease, and she wants to investigate it herself. She also knows that Bob has a database containing DNA patterns about various diseases. After Alice gets a sample of her DNA sequence, she sends it to Bob, who will then tell Alice the diagnosis. However, if Alice is concerned about her privacy, the above process is not acceptable because it does not prevent Bob from knowing Alice's

[*]Portions of this work were supported by Grant EIA-9903545 from the National Science Foundation, by sponsors of the Center for Education and Research in Information Assurance and Security at Purdue University, and by the Center for Computer Application and Software Engineering (CASE) at Syracuse University.
[†]In *Proceedings of New Security Paradigms Workshop*, pages 11-20, Cloudcroft, New Mexico, USA, September 11-13, 2001.

private information–both the DNA and the query result.

2. After a costly market research, company A decided that expanding its market share in some region will be very beneficial. However A is aware that another competing company B is also planing to expand its market share in some region. Strategically, A and B do not want to compete against each other in the same region, so they want to know whether their regions overlap without giving away location information (not only would disclosure of this information cost both companies a lot of money, it can also cause significant damage to the company if it is disclosed to other parties, e.g. another bigger competitor could then immediately occupy the market there before A or B even starts; or some real estate company could actually raise their price during the negotiation if they know A or B is very interested in that location). Therefore, they need a way to solve the problem while maintaining the privacy of their locations.

3. Two financial organizations plan to cooperatively work on a project for their mutual benefit. Each organization would like its own requirements being satisfied (usually, these requirements are modeled as linear equations or linear inequalities). However, their requirements are proprietary data that includes the customer's projects of the likely future evolution of certain commodity prices, interest and inflation rates, economic statistics, portfolio holdings. Therefore, nobody likes to disclose its requirements to the other party, or even to a "trusted" third party. How could they cooperate on this project while preserving the privacy of the individual information?

The common property of the above three examples is the following: two or more parties want to conduct a computation based on their private inputs, but neither party is willing to disclose its own input to anybody else. The problem is how to conduct such a computation while preserving the privacy of the inputs. This problem is referred to as Secure Multi-party Computation problem (SMC) in the literature [39]. Generally speaking, a secure multi-party computation problem deals with computing any probabilistic function on any input, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation than can be inferred from that participant's input and output [18].

Currently, to solve the above problems, a commonly strategy is to assume the trustworthiness of the service providers, or to assume the existence of a trusted third party, which is risky in nowadays' dynamic and malicious environment. Therefore protocols that can support joint computations while protecting the participants' privacy are of growing importance. In theory, the general secure multi-party computation problem is solvable [39, 31, 16] but, as Goldreich points out in [16], using the solutions derived by these general results for special cases of multi-party computation can be impractical; special solutions should be developed for special cases for efficiency reasons.

Goldwasser predicts that "the field of multi-party computations is today where public-key cryptography was ten years ago, namely an extremely powerful tool and rich theory whose real-life usage is at this time only beginning but will become in the future an integral part of our computing reality" [18].

Goldreich's observation and Goldwasser's prediction motivated us to search for specific SMC problems that have "real-life usage", as well as to search for their solutions. To this end, we have investigated, under the secure multi-party computation context, many specific computation domains, such as data mining, intrusion detection, database query, scientific computation, geometric computation, and statistical analysis. The results bring many interesting problems. The goal of this paper is to document the results of this research and present remaining open problems.

To search for new SMC problems systematically, we have proposed a transformation framework that allows us to systematically transform normal computations (not necessarily security related) to secure multi-party computations. Further research on these resultant problems reveals a number of interesting new problems. We will describe these new problems in this paper as well as discussing their potential applications and the related work, if any. It is important to point out that the list of problems presented in this paper is not intended to be an exhaustive list; we believe there are many other SMC problems in every specific computation domain. Our paper provides a framework and serves as a guidelines for researchers who work in other domains to define new SMC problems for their specific computations.

The framework in this paper has already triggered a number of interesting investigations. Some problems are currently under investigation, such as privacy-preserving cooperative statistical analysis [14], privacy-preserving cooperative scientific computations [13], and privacy-preserving geometric computation [4], privacy-preserving database query [12], and privacy-preserving intrusion detection.

2. RELATED WORK

The history of the multi-party computation problem is extensive since it was introduced by Yao [39] and extended by Goldreich, Micali, and Wigderson [31], and by many others. These works all use a similar methodology methodology: the computation problem is first represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. While this approach is appealing in its generality and simplicity, the protocols it generated depend on the size of the circuit. This size depends on the size of the input domain, and on the complexity of expressing such a computation.

In the past, secure multi-party computation research has mostly been focusing on theoretical studies, and few applied problems have been studied. In the literature, there are a few examples of secure multi-party computation problems, such as the Private Information Retrieval problem (PIR), privacy-preserving statistical database, and privacy-preserving data mining.

The PIR problem consists of a client and server; the client

needs to get the ith bit of a binary sequence from the server without letting the server know i; the server does not want the client to know the binary sequence either. A solution for this problem is not difficult; however an efficient solution, in particular a solution with small communication cost, is not easy. Studies [26, 8, 24, 23, 27, 30, 28, 19] have shown that one can design a protocol to solve the PIR problem with much better communication complexity than by using the general theoretical solutions.

The privacy-preserving data mining problem is another specific secure multi-party computation problem that has been discussed in the literature. Recently, two different privacy-preserving data mining problems were proposed by Lindell and Agrawal, respectively. In Lindell's paper [29], the problem is defined as this: Two parties, each having a private database, want to jointly conduct a data mining operation on the union of their two databases. How could these two parties accomplish this without disclosing their database to the other party, or any third party. In Agrawal's paper [1], the privacy-preserving data mining problem is defined as this: Alice is allowed to conduct data mining operation on a private database owned by Bob, how could Bob prevent Alice from accessing precise information in individual data records, while Alice is still able to conduct the data mining operations? The solution to these two similar problems are quite different: Lindell and Pinkas use secure multi-party computation protocols to solve their problem, while Agrawal uses the data perturbation method.

Apart from the above problems, secure multi-party computation problems exist in many other computation domains as well, and most of them have not been studied before. These new problems emerge if we combine the privacy concerns with the cooperative computation in a specific computation domain. The purpose of this paper is to document how we identify those problems, and the definition of them. We hope to motivate more people to look at these research problems. The authors of this paper have already studied some of these problems [12, 13, 14, 4].

Note that in some situations, only part of the data set needs to be kept confidential. For example, when two retail stores want to conduct a joint computation on their joint data, they are only concerned about their customers' names, not about each single transaction. In these cases, the problems could be solved using pseudonyms techniques [6, 5].

3. FRAMEWORK

We introduce a transformation framework that systematically transforms normal computations to secure multi-party computations. We start from describing two different models of computation (without the privacy requirements), and then we show how to transform them to models enhanced with privacy requirements, thus generating new SMC problems. The model after the transformation is the *Secure Multi-party Computation (SMC)* model.

According to the number of distinguished inputs, we classify computations into two different models: the multi-input computation model and the single-input computation model. The multi-input computation model usually has two distinguishable inputs. For instance, client/server computation is a multi-input computation model. The single-input computation model usually has one input or one set of inputs. For example, in data mining and statistical analysis, all the inputs usually come from one data set although the inputs consist of multiple data items.

Next we want to transform both models to the Secure Multi-party Computation model, in which, the input from each participating party is considered as private, and nobody is willing to disclose its own inputs to the other parties. In certain specific cases, the computation results could also be private, namely some party should not learn the results.

For the multi-input computation model, its transformation to the corresponding SMC model is straightforward because the model naturally has at least two inputs. Therefore, if we treat each input as coming from a different party, the new problem now becomes "how to conduct the same computation while maintaining the privacy of each party's input". Figure 1(a) demonstrates such a transformation.

For the single-input computation model, since it only has one input, we cannot use the same transformation as we used for the multi-input computation model; we have to somehow transform the model to a multi-input computation model. Let us call this computation C, and assume that the single input is a set D of data items. If we can divide D into two disjoint data set D_1 and D_2, we will have a multiple-input computation model. There are many ways to divide D into two data sets, and each way could lead to a different SMC problem. We are focusing on two types of transformations: homogeneous transformation and heterogeneous transformation.

In the homogeneous transformation, D's data items are divided to two sets, but each single data item is not cut into two parts. For example, if D is a database of student records, the homogeneous transformation will put a subset of the records into one data set, and the rest of the the records into another data set; however, each student's record is not cut into two parts. In other words, the two generated data sets maintain the same set of features. Figure 1(b) demonstrates such a transformation.

In the heterogeneous transformation, each single data item is cut into two parts, with each part going to a separate data set. Taking the same example used above, if each student record contains a student's academic record and medical record, the heterogeneous transformation could put all students' academic records into one data set, and all students' medical records into another data set. In other words, the two generated data sets maintain different set of features. Figure 1(c) demonstrates such a transformation.

After the above transformation, the new problem now becomes "how to conduct the computation C on the union of D_1 and D_2, where D_1 belongs to one party and D_2 belongs to another party, and neither of these two parties wants to disclose his or her private data set to other.

Privacy Requirements

To decide whether a solution achieves the privacy requirements, we need to know the formal definition of *privacy*.

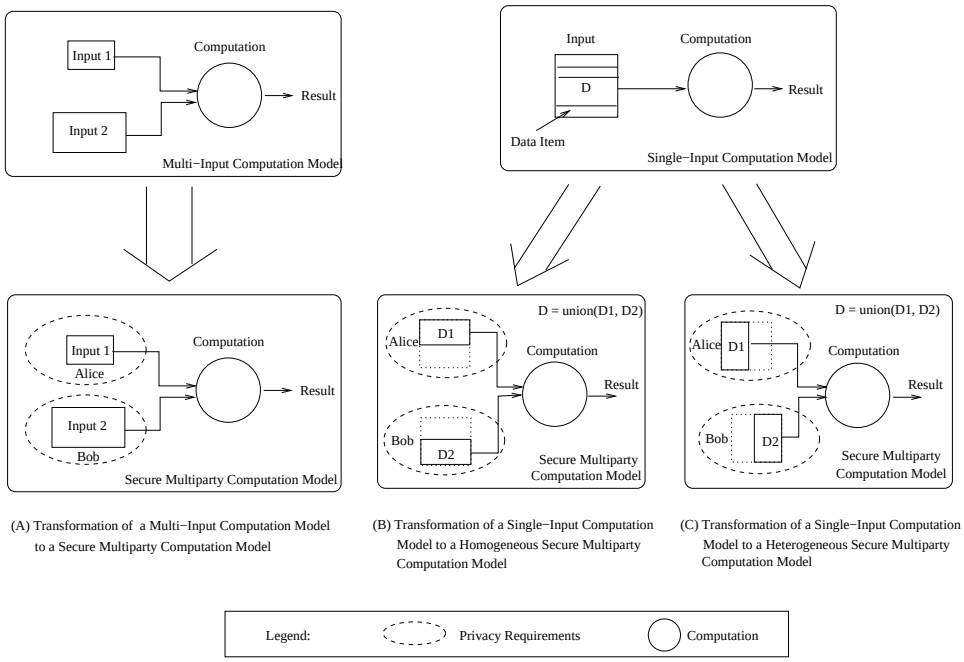

Figure 1: Transformation Framework

Goldreich has provided a formal definition of privacy in [16], and has provided a solid theoretical background that solutions to a specific secure multi-party computation problem should base on. Please refer to [16] for the details.

4. SPECIFIC SECURE MULTI-PARTY COMPUTATION PROBLEMS

In this section, we will investigate a number of specific computations including database query, intrusion detection, data mining, geometric computation, statistical analysis, scientific computation, and some miscellaneous computations. For each computation, we will apply the transformation framework to transform it to a Secure Multi-party Computation problem. As will be shown, some of the resulting problems are new, and some have already been under study in the past. For the known problems, we give a brief survey on the related work; for the new problems, we describe their potential real-life applications. We also transform some well known computation problems to corresponding SMC problems whose applications are yet unknown at this time, but because of the original problem is so useful in the real life that we believe the corresponding SMC problem will eventually be applicable.

We emphasize that the list presented in this section is not intended to be exhaustive; we conjecture that many more new research problems could arise following the models described in this paper. This section serves as a guideline with examples for those researchers who work in their specific domain to define new SMC problems for their specific computations.

4.1 Privacy-Preserving Cooperative Scientific Computations

PROBLEM 1. *(Linear Systems of Equations) Alice has m private linear equations represented by $M_1x = b_1$, and Bob has $n-m$ private linear equations represented by $M_2x = b_2$, where x is an n-dimensional vector. Alice and Bob want to find a vector x that satisfies both of Alice's and Bob's equations.*

PROBLEM 2. *(Linear Least Squares Problem) Alice has m_1 private linear equations represented by $M_1x = b_1$, and Bob has m_2 private linear equations represented by $M_2x = b_2$, where x is an n-dimensional vector and $m_1 + m_2 > n$. Alice and Bob want to find a vector x that satisfies both of Alice's and Bob's equations. Since there are more conditions (equations) to be satisfied than degrees of freedom (variables), it is unlikely that they can all be satisfied. Therefore, they want to attempt to satisfy the equations as best as they can–that is, make the size of the residual vector r with components*

$$r_j = b_j - \sum_{i=1}^{n} a_{ji}x_i$$

as small as possible (a_{ji} are the entries in the new matrix formed from M_1 and M_2). The least-squares criterion is the use of the Euclidean (or least-squares) norm for the size of r; that is, minimize

$$\sqrt{\sum_{j=1}^{m_1+m_2} r_j^2} = \| r \|_2$$

PROBLEM 3. *(Linear Programming) Alice has m_1 private linear requirements represented by $M_1x \leq b_1$, and Bob has another m_2 private linear requirements represented by $M_2x \leq b_2$, where x is an n-dimensional vector. They want to minimize (maximize) the value of $a_1 * x_1 + \cdots + a_n * x_n$, for the*

known $a_1, \ldots, a_n$, and the solution $x = (x_1, \ldots, x_n)$ should satisfy all of Alice's and Bob's requirements.

The linear systems of equations problem, the linear least squares problem and the linear programming problem have proved valuable for modeling many and diverse types of problems in planning, routing, scheduling, assignment, and design. Industries that make use of these problems and their extensions include transportation, energy, telecommunications, and manufacturing of many kinds. In many cases, those linear equations or linear requirements are proprietary data and are too valuable to disclose to anybody else, especially to a potential competitor.

For instance, in one of the examples mentioned in the beginning of this paper, two financial organizations plan to cooperatively work on a project for mutual benefit. Each of the organizations would like its own requirements being satisfied (usually, these requirements are modeled as linear equations or linear inequalities). However, most of their requirements are very likely their proprietary data which includes the customer's projects of the likely future evolution of certain commodity prices, interest and inflation rates, economic statistics, portfolio holding, etc. Therefore, nobody likes to disclose its requirements to the other party, or even to a "trusted" third party. How could they cooperate on this project that has to satisfy everybody's private requirements without compromising the privacy requirements? The current practice is to operate "in the clear", that is, by revealing requirements to the other party or to the agent performing the computation. The consequence is obvious if the other party or the agent is not trusted. The solutions to the above problems actually provide a secure way to solve this problem.

We have recently proposed protocols to solve the above three problems in [13].

4.2 Privacy-Preserving Database Query

PROBLEM 4. *(Database Query)* Alice has a string q, and Bob has a database of strings $T = \{t_1, \ldots, t_N\}$; Alice wants to know whether there exists a string t_i in Bob's database that "matches" q. The "match" could be an exact match or an approximate (closest) match. The privacy requirement is that Bob cannot know Alice's secret query q or the response to that query, and Alice cannot know Bob's database contents except for what could be derived from the query result.

The exact matching problem has been extensively considered in the literature [26, 8, 24, 23, 27, 30, 28, 19], even though it can theoretically be solved using the general techniques of secure multi-party computation [16]. The motivation for giving these specialized solutions to it is that they are more *efficient* than those that follow from the abovementioned general techniques. This is also our motivation for studying approximate pattern matching even though it too is a special case of the general secure multi-party computation problem. Unlike exact pattern matching that produces "yes" and "no" answers, approximate pattern matching measures the difference between the two targets, and produces a *score* to indicate how different the two targets are. The metrics used to measure the difference usually are heuristic and are application-dependent. For example, in image template matching [20, 25], $\sum_{i=1}^{n}(a_i - b_i)^2$ and $\sum_{i=1}^{n}|a_i - b_i|$ are often used to measure the difference between two sequences a and b. In DNA sequence matching [22], *edit distance* [3, 10] makes more sense than the above measurements; *edit distance* measures the cost of transforming one given sequence to another given sequence, and its special case, *longest common subsequence* is used to measure how similar two sequences are.

Solving approximate pattern matching problems under such privacy constraints is quite a nontrivial task. Consider the $\sum_{i=1}^{n}|a_i - b_i|$ metric as an example. The known PIR (private information retrieval) techniques [26, 8, 24, 23, 27, 30, 28, 19] can be used by Alice to efficiently access each individual b_i without revealing to Bob anything about which b_i Alice accessed, but doing this for each individual b_i and then calculating $\sum_{i=1}^{n}|a_i - b_i|$ violates the requirement that Alice should know the total score $\sum_{i=1}^{n}|a_i - b_i|$ *without knowing anything other than that score*, i.e., without learning anything about the individual b_i values. Using a general secure multi-party computation protocol typically does not lead to an efficient solution. The goals of this research, is to find efficient ways to do such approximate pattern matchings without disclosing private information.

We have published some research results regarding to this problem in [12].

4.3 Privacy-Preserving Intrusion Detection

PROBLEM 5. *(Profile Matching) Alice has a profile database containing many known hacker's behaviors; Bob has collected a hacker's behavior from a recent break-in, and he wants to identify the hacker by matching this hacker's behavior with Alice's profile database. However, Bob doesn't want to disclose the hacker's actual behavior to Alice because that might disclose the vulnerability in his system because that behavior could be a successful series of actions that leads to the compromise of his system. On the other hand, Alice doesn't want to disclose the profile database because of the database contains confidential information. How could Alice and Bob cooperatively accomplish this task without sacrificing their privacy?*

PROBLEM 6. *Two major financial organizations wants to cooperate in preventing fraudulent intrusion into their computing system. To this end, they need to share data patterns relevant to fraudulent intrusion, but they do not want to share the data patterns since they are sensitive information. Therefore, combining the databases is not feasible. How can these two financial organizations conduct data mining operation or machine learning operation on the joint of their data while maintaining the privacy of the data.*

In nowadays, many major banks share information fairly freely in cooperative intrusion investigations, but they have to be careful because of lawsuits. The solutions to the above problems could prevent the banks from getting into the legal troubles.

4.4 Privacy-Preserving Data Mining

PROBLEM 7. *(Classification) Alice has a private structured database D_1, and Bob has another private structured database D_2; both of the structured database are comprised of attribute-value pairs. Each row of the database is a transaction and each column is an attribute taking on different values. One of the attributes in the database is designated as the class attribute. How could Alice and Bob build a decision tree based on the $D_1 \cup D_2$ without disclosing the content of the database to the other party?*

Given a decision tree, one can predict the class of new transactions for which the class is unknown. There are several proposed algorithms for generating decision trees; however, if the database D_1 or D_2 should be kept private from anybody other than the owner, those algorithms does not work because a default assumption for those algorithms is that the whole database is available. A new algorithm is needed to solve this new problem.

ID3 algorithm is one of the proposed algorithms for generating decision trees. Based on the ID3 algorithm, Lindell and Pinkas proposed a solution to the above privacy-preserving classification problem in [29] using secure multi-party computation protocol.

PROBLEM 8. *(Data Clustering) Alice has a private database D_1, and Bob has a private database D_2. They want to jointly perform data clustering on the union of D_1 and D_2.*

Basically, data clustering is to group a set of data (without a predefined class attribute), based on the conceptual clustering principle: *maximizing the intraclass similarity and minimizing the interclass similarity.*

PROBLEM 9. *(Mining Association Rules) Alice has a private database D_1, and Bob has a private database D_2. They want to jointly identify association rules in the union of D_1 and D_2.*

For example, country A's intelligence agents have observed the activities $X = (x_1, ..., x_n)$ for a period of time, and Country B's intelligence agents have observed the activities $Y = (y_1, ..., y_m)$ for the same period of time. They want to collaboratively find out whether the activities in Y has any correlation with the activities in X. The results of collaboration could help both countries to understand the trend of the behaviors of the target, such as the behaviors of some suspected terrorism organization, the military movement of a dangerous country, etc. However, neither A or B is willing to disclose its observation to the other countries because they don't fully trust each other. It is possible that B might use A's intelligence information (or sell it to the target) to uncover A's agents, and thus causing damage to A's intelligence agents.

PROBLEM 10. *(Data Generalization, Summarization and Characterization) Alice has a private database D_1, and Bob has a private database D_2. They want to generalize, summarize or characterize the union of these two database.*

The above privacy-preserving data mining problems are related to another research problem–Distributed Data Mining (DDM) problem. Distributed data mining [2] is a fast growing area that deals with the problem of finding data patterns in an environment with distributed data and computation. A good DDM algorithm analyzes data in a distributed fashion with modest data communication overhead. Typically DDM algorithms involve local data analysis followed by the generation of a global data model through the aggregation of the local results; therefore, it preserve the privacy of the local data to some extent, but, the global data model generated locally might still contain sensitive information that they do not want to disclose.

We believe, results from the distributed data mining field will be helpful in solving the privacy-preserving data mining problem.

4.5 Privacy-Preserving Geometric Computation

PROBLEM 11. *(Intersection) Alice has a private shape a, and Bob has another private shape b; they both want to know whether a and b intersect? Alice does not want Bob or anybody else know any information about the shape a, nor does Bob want to disclose information about his shape b. Moreover, in no case should anybody learn the relative position between a and b, and, if these two regions intersect, nobody should learn where they intersect with each other.*

Much of the motivation for studying the privacy-preserving intersection problem stems from the simple fact that two private objects cannot occupy the same place at the same time. For example, in the beginning of the paper, we have described an example in which two companies plan to expand their market shares in certain regions, but, they do not want to compete in the same region. What they really want to know is whether their selected regions overlap with each other. Because the information about its own selected region is so valuable to each company, neither of them wants to disclose it to the other party.

PROBLEM 12. *(Point-Inclusion) Alice has a private point z, and Bob has a private polygon P. They want to find out whether the point is inside the polygon or not? Alice does not want Bob or anybody else to know any information about the point, and likewise, Bob does not want anybody else to know any information about the polygon. Furthermore, Alice and Bob can only learn whether the point is inside or outside of the polygon, nobody is allowed to learn the relative position between the point and the polygon, such as whether z is at the northwest side of P, or whether z is close to one of the border of the polygon, and so on.*

This problem is very useful in the many scenarios, such as the following: country A decides to bomb a location x in another country, but A does not want to hurt its relationship with its friends, who might have some areas of interests in the bombing region. Those countries might have secret businesses, secret military bases, or secret agencies in that area. Obviously, A does not want to disclose the exact location

of x to all of its friends, except the one that will definitely be hurt by this bombing; on the other hand, its friends do not want to disclose their secret areas to A either, unless they are in the target area. If the target is not within the secret areas, no information should be disclosed, including the information such as whether the target is at the west of the area, or within certain longitude or latitude. Basically it is "all-or-nothing": if one will be bombed, it knows all; otherwise it knows nothing. How could they solve this dilemma?

PROBLEM 13. *(Range Searching) Alice has a private range (represented by either a hyper-rectangular shape or by spherical shape), and Bob has N private points. Alice and Bob want to jointly find out the number of points in Alice's range; however, neither of them is willing to disclose their data to the other party.*

Range searching arises in a wide range of applications, including geographic information systems, spatial database, and time-series database. In many cases, both the query and the database contain confidential information; therefore, to provide the range query service, solutions to the privacy-preserving range searching problem are needed.

PROBLEM 14. *(Closest Pair) Alice has M private points in the plane, Bob has N private points in the plane. Alice and Bob want to jointly find two points among these $M + N$ points, such that their mutual distance is smallest.*

PROBLEM 15. *(Convex Hulls) Alice has M private points in the plane, Bob has another N private points in the plane. Alice and Bob want to jointly find the convex hulls for these $M + N$ points; however, neither Alice nor Bob wants to disclose any more information to the other party than what could be derived from the result.*

The authors of this paper have recently proposed solutions to the point-inclusion problem and the intersection problem in [4].

4.6 Privacy-Preserving Statistical Analysis

PROBLEM 16. *(Correlation and Regression Analysis) Alice has a private data set $D_1 = (x_1, \ldots, x_n)$, Bob has another private data set $D_2 = (y_1, \ldots, y_n)$, where x_i is the value of variable x, and y_i is the corresponding value of variable y. Alice and Bob want to find out the following results without compromising the privacy of their data set:*

1. *correlation coefficient between x and y: the strength of a linear relationship between x and y, namely the degree to which larger x values go with larger y values and smaller x values go with smaller y values.*
2. *regression line: an equation that provides values of y for given value of x. The objectives of regression analysis is to make predictions.*

This problem has a lot of applications. For example, a bank wants to investigate if ages can affect people's financial activities. However, the bank only has customers' financial activities, it does not know the ages of its customers. Therefore, the bank turns to some government bureau who has the knowledge of every person's dates of birth, but the government bureau is required by laws not to disclose it. On the other hand, the customers' financial activities are the bank's proprietary data that the bank does not want to disclose to anybody. The solutions to the privacy-preserving statistical analysis problem could be used to solve this problem.

In another example, a school wants to investigate the relationship between people's intelligence quotient (IQ) score and their annual salary. The school has its students' IQ score, but does not have students' salary information; therefore the school needs to cooperate with companies that hire the students, but those companies are not willing to disclose the salary information. On the other hand, the school cannot give students' IQ score to their employers either. A privacy-preserving statistical analysis method [14] is needed to solve this problem.

Some other privacy-preserving statistical analysis problems and solutions have been proposed in the statistics community. For example, Random response techniques were proposed [37, 38, 21, 33] to compute the mean value of a sample data without knowing the actual sample data.

4.7 Other Specific Secure Multi-Party Computation Problems

There are many other interesting secure multi-party computation problems, but here we will only describe some of them without discussing their applications. For some of them, their applications are obvious, but for some others, their real life applications are yet unknown. We will leave those to readers to justify whether they are useful, we also hope this can trigger readers to think about more useful problems in some other specific computation domains.

1. *Selection problem* (select median, select the kth smallest element): Alice has a private data set d_1, and Bob has a private data set d_2; they want to find the median (or the kth smallest element) among the data in $d_1 \cup d_2$.
2. *Sorting problem*: Alice has a private data set d_1, and Bob has a private data set d_2; they want to sort the elements in the union of these two data sets, such that each element in these two data sets is marked by a number representing the order of this element.
3. *Shortest path problem*: Alice and Bob each has a private graph represented by $g1$ and $g2$, respectively, and the links between these two graphs are known to both of them. Given any two points (they could be in a same graph, or in different graphs), how could Alice and Bob jointly compute the shortest distance (or path) between these two points. One of the applications of this problem is the network traffic routing between two private network service providers if they do not want to disclose too much information about their own private network. We already had a solution to this problem.

4. *Privacy-Preserving polynomial interpolation*: Alice has n_1 private pairs (x_i, y_i), for $i = 0, \ldots, n_1$, Bob has n_2 private pairs (x_j, y_j), for $j = n_1 + 1, \ldots, n$. Suppose $x_0, \ldots, x_n$ are distinct data points, how can Alice and Bob jointly find the polynomial $p(x)$ of degree n that interpolates the data set $\{(x_0, y_0), \ldots, (x_n, y_n)\}$, i.e. $p(x_k) = y_k$ for all $k = 0, \ldots, n$.

5. OUTLINE OF SOME APPROACHES

While each SMC problem in a specific domain need a specific solution, there are certain general approaches that we have adopted to solve the new SMC problems. These approaches are based on many cryptographic tools including zero-knowledge proof [35], oblivious transfer [34], 1-out-of-n oblivious transfer [17, 9] oblivious evaluation of polynomials [32], secret sharing [36], threshold cryptography [15, 11], Yao's Millionaire Protocol [39, 7]. We will only give an overview of the approaches that we used in solving some of the problems described in this paper because the main purpose of this paper is to present the set of new problems, rather than the specific techniques in solving them.

As we know, to solve a cooperative computation problem (in the normal case), one party, Alice, can send her inputs to the other party, Bob, who can then solve the computation problem by himself. This naive solution is not good in the privacy-preserving context because Bob can immediately find out Alice's inputs. However, what we have learned from this naive solution is that if Alice can somehow send her inputs to Bob in such a way that makes it impossible for Bob to derive Alice's input while still allowing Bob to solve the problem by himself, then we do not need to worry about how Bob solve the problem, because now Bob has all the inputs, he should be able to solve the problem by himself. Therefore, to use this approach, the most important step is to send Alice's inputs to Bob while preserving Alice's privacy.

For example, we used the above approach to solve Problem 1, the linear privacy-preserving linear systems of equations problem. Actually, we solved the following more general problem: *Alice has a private matrix M_1 and a private vector b_1, and Bob has a private matrix M_2 and a private vector b_2, where M_1 and M_2 are $n \times n$ matrices, and b_1 and b_2 are n-dimensional vectors. Without disclosing their private inputs to the other party, Alice and Bob want to solve the linear equation:* $(M_1 + M_2)x = b_1 + b_2$.

Our solution is based on the fact that the solution to the linear equations $(M_1+M_2)x = b_1+b_2$ is equivalent to the solution to the linear equations $P(M_1+M_2)QQ^{-1}x = P(b_1+b_2)$. If Bob knows $M' = P(M_1 + M_2)Q$ and $b' = P(b_1 + b_2)$, he can solve the linear equation problem: $M'\hat{x} = b'$, and thus getting the final solution x, where $x = Q\hat{x}$. But how could Bob know M' and b' without being able to derive the value of M_1 and b_1? To solve this problem, Alice generates two random $n \times n$ matrices P and Q with Q being invertible. With the help of 1-out-of-N Oblivious Transfer protocol [17, 9], Bob is able to learn the value of $P(M_1 + M_2)Q$ and $P(b_1+b_2)$. However, Bob will not learn the value of PM_1Q, PM_2Q, Pb_1, or Pb_2, much less P, Q, M_1, or b_1. After Bob gets $M' = PM_1Q+PM_2Q$ and $b' = Pb_1+Pb_2$, he can solve the linear equations $M'\hat{x} = b'$, and then send the solution $\hat{x}$ to Alice, who can compute the final solution $x = Q\hat{x}$. Finally Alice sends the solution to Bob. The complete solution of this problem is described in [13]

As we mentioned that the general secure multi-party computation solution (the circuit evaluation) is not practical because the protocols it generates depend on the size of the input domain, and on the complexity of expressing the computation as a circuit (for example, a multiplication circuit is quadratic in the size of its inputs). However, if the size of the input domain and the complexity of the circuit are small, the general solution could be practical. Based on this observation, one approach to solve a specific secure multi-party computation problem is to reduce the problem to sub-problems that have small input domains and small circuit size.

For example, to solve the Intersection problem (Problem 11), we reduced the problem to three sub-problems: 1) evaluation of a linear function problem 2) comparison problem and 3) evaluation of a boolean expression problem. The first two problems can be solved using known techniques proposed in the literature, in particular, the oblivious polynomial evaluation protocol [32] and Yao's millionaire protocol [39, 7]. The third problem could be solved using the general secure multi-party computation solution, namely the circuit evaluation protocol. Because in this sub-problem, the input domain is just $\{0, 1\}$–a very small domain, and the complexity of building a circuit to evaluate a boolean expression is just linear to the number of items in this boolean expression. The complete solution of this problem is described in [4]

Sometimes, it helps to find an efficient solution by introducing into the protocol a third party–an untrusted third party. This third party should not learn anything about either participant's inputs. This approach has been used in the literature to solve some secure multi-party computation problems. For example, to solve Yao's Millionaire problem [39], Cachin uses an untrusted third party, and has significantly improved the performance of the solution to the problem.

6. CONCLUSION AND FUTURE WORK

In this paper, we have studied several specific computations, such as database query, intrusion detection, data mining, geometric computation, statistical analysis, and scientific computations. We studies these computations from another perspective–secure multi-party computation perspective, i.e. how to conduct these computations among multiple parties while maintaining the privacy of each party's input. As results, we have defined a number of secure multi-party computation problems, among which some are well studied for decades, some are studied in recent years, and some are just new problems.

We have only studied a limited number of computations domains in this paper, because it is not our intention to provide a complete lists of new SMC problems. We want to provide a guideline for the researchers in other computation areas to think about their computation problem from this security perspective, thus coming up with new SMC problems, if necessary.

Among those problems list in the paper, some are not solved yet, some are under active research, and some have triggered interests from people who works on a variety of computation domains. We hope that after working on several problems we can gain more insights on how to solve this type of problems, what the useful building blocks for solving this type of the problem are, how the solutions to the existing problem (without the privacy requirements) could help to solve those specific secure multi-party computation problems.

7. ACKNOWLEDGMENTS

We would like to thank the participants in the New Security Paradigms Workshop for their helpful comments. A special thank goes to Bob Blakley, for providing excellent notes of the discussion.

8. REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD on Management of data*, pages 439–450, Dallas, TX USA, May 15 - 18 2000.

[2] H. Kargupta, B. Park, D. Hershberger and, E. Johnson. Collective data mining: A new perspective toward distributed data mining. *Advances in Distributed and Parallel Knowledge Discovery*, 1999.

[3] A. Apostolico and Z. Galil, editors. *Pattern Matching Algorithms*. Oxford University Press, 1997.

[4] Mikhail J. Atallah and Wenliang Du. Secure multi-party computational geometry. In *WADS2001: Seventh International Workshop on Algorithms and Data Structures*, pages 165–179, Providence, Rhode Island, USA, August 8-10 2001.

[5] J. Biskup and U. Flegel. On pseudonymization of audit data for intrusion detection. In *Workshop on Design Issues in Anonymity and Unobservability*, pages 161–180, 2000.

[6] J. Biskup and U. Flegel. Transaction-based pseudonyms in audit data for privacy respecting intrusion detection. In *Recent Advances in Intrusion Detection*, pages 28–48, 2000.

[7] C. Cachin. Efficient private bidding and auctions with an oblivious third party. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 120–127, Singapore, November 1-4 1999.

[8] B. Chor and N. Gilboa. Computationally private information retrieval (extended abstract). In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, El Paso, TX USA, May 4-6 1997.

[9] G. Brassard, C. Crépeau and J. Robert. All-or-nothing disclosure of secrets. In *Advances in Cryptology - Crypto86, Lecture Notes in Computer Science*, volume 234-238, 1987.

[10] M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, 1994.

[11] Y. Desmedt. Some recent research aspects of threshold cryptography. In *Lecture Notes in Computer Science 1396*, pages 158–173. Springer-Verlag, 1997.

[12] Wenliang Du and Mikhail J. Atallah. Protocols for secure remote database access with approximate matching. In *7th ACM Conference on Computer and Communications Security (ACMCCS 2000), The First Workshop on Security and Privacy in E-Commerce*, Athens, Greece, November 1-4 2000.

[13] Wenliang Du and Mikhail J. Atallah. Privacy-preserving cooperative scientific computations. In *14th IEEE Computer Security Foundations Workshop*, pages 273–282, Nova Scotia, Canada, June 11-13 2001.

[14] Wenliang Du and Mikhail J. Atallah. Privacy-preserving statistical analysis. In *Proceedings of the 17th Annual Computer Security Applications Conference*, pages 102–110, New Orleans, Louisiana, USA, December 10-14 2001.

[15] P. Gemmell. An introduction to threshold cryptography. In *CryptoBytes*, volume 2. RSA Laboratories, 1997.

[16] O. Goldreich. Secure multi-party computation (working draft). Available from http://www.wisdom.weizmann.ac.il/home/oded/public_html/foc.html, 1998.

[17] S. Even, O. Goldreich and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28:637–647, 1985.

[18] S. Goldwasser. Multi-party computations: Past and present. In *Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing*, Santa Barbara, CA USA, August 21-24 1997.

[19] Y. Gertner, S. Goldwasser and T. Malkin. A random server model for private information retrieval. In *2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM '98)*, 1998.

[20] R. Gonzalezi and R. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.

[21] M. S. Goodstadt and V.Gruson. The randomized response technique: A test on drug use. *Journal of the American Statistical Association*, 70(352):814–818, December 1975.

[22] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Comutational Biology*. Cambridge University Press, 1997.

[23] G. Di-Crescenzo, Y. Ishai and R. Ostrovsky. Universal service-providers for database private information retrieval. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*, September 21 1998.

[24] Y. Ishai and E. Kushilevitz. Improved upper bounds on information-theoretic private information retrieval (extended abstract). In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, Atlanta, GA USA, May 1-4 1999.

[25] A. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.

[26] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan. Private information retrieval. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, Milwaukee, WI USA, October 23-25 1995.

[27] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proceedings of the 38th annual IEEE computer society conference on Foundation of Computer Science*, Miami Beach, Florida USA, October 20-22 1997.

[28] Y. Gertner, Y. Ishai, E. Kushilevitz and T. Malkin. Protecting data privacy in private information retrieval schemes. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, Dallas, TX USA, May 24-26 1998.

[29] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Advances in Cryptology - Crypto2000, Lecture Notes in Computer Science*, volume 1880, 2000.

[30] C. Cachin, S. Micali and M. Stadler. Computationally private information retrieval with polylogarithmic communication. *Advances in Cryptology: EUROCRYPT '99, Lecture Notes in Computer Science*, 1592:402–414, 1999.

[31] O. Goldreich, S. Micali and A. Wigderson. How to play any mental game. In *Proceedings of the 19th annual ACM symposium on Theory of computing*, pages 218–229, 1987.

[32] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation (extended abstract). In *Proceedings of the 31th ACM Symposium on Theory of Computing*, pages 245–254, Atanta, GA, USA, May 1-4 1999.

[33] K. H. Pollock and Y. Bek. A comparison of three randomized response models for quantitative data. *Journal of the American Statistical Association*, 71(356):994–886, December 1976.

[34] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report Tech. Memo TR-81, Aiken Computation Laboratory, 1981.

[35] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., 1996.

[36] A. Shamir. How to share a secret. *Communication of the ACM*, 22(11):612–613, 1979.

[37] S. L. Warner. Randomized response: A surey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, March 1965.

[38] S. L. Warner. Randomized response: A surey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 66(336):884–888, December 1971.

[39] A.C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 1982.