Secure Multiparty Overall Mean Computation via Oblivious Polynomial Evaluation

Mert Özarar\textsuperscript{1} and Attila Özgit\textsuperscript{1}

\textsuperscript{1} Department of Computer Engineering, Middle East Technical University, 06531, Ankara, Turkey  
\{ozarar, ozgit\}@ceng.metu.edu.tr

Abstract. The number of opportunities for cooperative computation has exponentially been increasing with growing interaction via Internet technologies. Most of the time, the communicating parties may not want to disclose their private data to the other principal while taking the advantage of collaboration, hence concentrating on the results rather than private and perhaps useless data values. To conduct such a computation while preserving the privacy of the inputs for a target case is referred to as Secure Multiparty Computation problem. In this work, the privacy preserving overall mean computation problem is analyzed. We present two protocols for two-party and multi-party case via oblivious polynomial evaluation is used as a cryptographic mechanism to assure the security of the private data. Under given security assumptions, the privacy validity of the algorithms are justified.

Keywords: Secure Multiparty Computation, Overall Mean, Oblivious Polynomial Evaluation

1 Introduction

The number of opportunities for cooperative computation has exponentially been increasing with growing interaction via Internet technologies. These computations could occur between trusted partners, between partially trusted partners, or even between competitors. Most of the time, the communicating parties may not want to disclose their private data to the other principal while taking the advantage of collaboration, hence concentrating on the results rather than private and perhaps useless data values. For example, two or more competing large organizations might jointly invest in a project that must satisfy all organizations' goals while preserving their private and valuable data [4]. For performing such computations, one party must know inputs from all the participants; however if none of the parties can be trusted enough to know all the inputs, privacy will become a primary concern. Hence the techniques for secure multiparty computation are quite relevant and practical to overcome the privacy gaps.
1.1 Privacy Constraints

The definition of privacy in the cryptographic community limits the information that is leaked by the distributed computation to be the information that can be learned from the designated output of the computation [15]. Although there are several variants of the definition of privacy, for the purpose of this discussion we use the definition that compares the result of the actual computation to that of an “ideal” computation. Consider a party that is involved in the actual computation of a function (e.g. a data mining algorithm). Consider also an “ideal scenario”, where there is also a “trusted party” who does not deviate from the behavior that we prescribe for him, and does not attempt to cheat. In the ideal scenario all parties send their inputs to the trusted party, who then computes the function with complete input data set and sends the results to the parties of the computation.

A protocol is secure if anything that an adversary can learn in the actual world, it can also learn in the ideal world, namely from its own input and from the output it receives from the trusted party [16]. In essence, this means that the protocol run for computing the function does not leak any “unnecessary” information. Of course, there are partial leaks of information that may be considered as harmless. It is hard, however, to decide which type of leakage can be tolerated. The cryptographic community therefore aims at designing protocols that do not reveal any information except for their designated output, and in many cases such protocols can be efficiently constructed.

1.2 Secure Multiparty Computation

If multiple parties want to perform a computation based on their private inputs, but neither party is willing to disclose its own input to anybody else, then the basic problem is how to conduct such a computation while preserving the privacy of the inputs. This is referred to as Secure Multiparty Computation problem (SMC) in the literature.

In general, a secure multiparty 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 other than that can be inferred from the participant's input and output [6]. Consider a trusted party who collects all participants’ data and then performs the computation and sends the results to the participants. Without having a trusted party, some communication among the participants is certainly required for any related computation; yet we do not know how to ensure that this communication doesn't disclose anything. One solution is to allow non-determinism in the exact values sent for the intermediate communication (e.g., encrypt with a randomly chosen key) and demonstrate that a party with just its own input and the result can generate a predicted intermediate computation that is as likely as the actual values. A detailed discussion of Secure Multiparty Computation is given in [5, 10].
1.3 Adversarial Behavior

Privacy preserving algorithms are designed in order to preserve privacy even in the presence of adversarial participants that attempt to gather information about the inputs of their peers. There are, however, different levels of adversarial behavior. Cryptographic research typically considers two types of adversaries: A semi-honest adversary (also known as a passive, or honest but curious adversary) is a party that correctly follows the protocol specification, yet attempts to learn additional information by analyzing the messages received during the protocol execution [16]. On the other hand, a malicious adversary (active) may arbitrarily deviate from the protocol specification. For example, consider a step in the protocol where one of the parties is required to choose a random number and broadcast it. If the party is semi-honest then we can assume that this number is indeed random. On the other hand, if the party is malicious, then he might choose the number in a sophisticated way that enables him to gain additional information.

It is of course easier to design a solution that is secured against semi-honest adversaries, than for malicious adversaries. A common approach is therefore, first to design a secure protocol for the semi-honest case, and then transform it into a protocol that is secure against malicious adversaries. This transformation can be done by requiring each party to use zero-knowledge proofs to justify that each step that it is taking follows the specification of the protocol. More efficient transformations are often required, since this generic approach might be rather inefficient and add considerable overhead to each step of the protocol. It is remarkable that the semi-honest adversarial model is often a realistic one. This is because deviating from a specified protocol which may be buried in a complex application is a non-trivial task, and because a semi-honest adversarial behavior can model a scenario in which the parties that participate in the protocol are honest, but following the protocol execution an adversary may obtain a transcript of the protocol execution by breaking into a machine used by one of the participants.

Based on the given background, we studied the secure multiparty statistical overall mean computation problem using oblivious polynomial evaluation (OPE) technique developed by [14]. A solution for a similar problem called “weighted average problem” (WAP) has already been proposed by [11] yet the “overall mean problem” (OMP) is a little bit more complicated than WAP due to the bilinear terms in the resultant functional. We believe that our contribution will demonstrate how the new paradigm can lead to practical solutions for applications involving statistical overall mean computation especially in the areas of data mining and machine learning.

The organization of the paper is as follows: Section 2 discusses the related work on SMC. Section 3 gives the OPE technique together with OMP. Then section 4 which is the heart of the paper gives the solution to privacy preserving OMP for two-party case. The extension to multiparty form is discussed in fifth section. Finally section 6 concludes this paper and proposes several future research directions.
2 Related Work

The basics of the secure multiparty computation problem is extensive since it was introduced by [17] and [18], then extended by [9] and by many others. 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 generated protocols 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.

The usage of SMC in privacy preserving applications is presented in [14] by the introduction of “oblivious transfer”. Oblivious transfer is a basic protocol that is the main building block of secure computation. It might seem strange at first, but its role in secure computation should become clear later. It was shown by [9] that oblivious transfer is sufficient for secure computation in the sense that given an implementation of oblivious transfer, and no other cryptographic primitive, one could construct any secure computation protocol.

The oblivious transfer protocol involves two parties, the sender and the receiver. The sender’s input is a pair \((x_0, x_1)\) and the receiver’s input is a bit \(\sigma \in \{0, 1\}\). At the end of the protocol the receiver learns \(x_\sigma\) (and nothing else) and the sender learns nothing. Oblivious transfer is often the most computationally intensive operation of secure protocols, and is repeated many times. Each invocation of oblivious transfer typically requires a constant number of invocations of trapdoor permutations (i.e. public-key operations, or exponentiations). It is possible to reduce the amortized overhead of oblivious transfer to one exponentiation per a logarithmic number of oblivious transfers, even for the case of malicious adversaries [16]. Oblivious polynomial evaluation is a technique based on oblivious transfer and explained in the next section.

There are a number of SMC applications distributed in broad range areas. Atallah et. al. [1] proposes preliminary work for solving computational geometry problems including scalar product, permutation, vector dominance, equality testing, point inclusion and intersection. In [4], the authors define a set of new privacy preserving cooperative scientific computation problems: privacy preserving cooperative linear system of equations problem and privacy preserving cooperative linear least-square problem. They have developed protocols to solve these problems. Du et. al. [3] also studies the problem of how to conduct the statistical analysis in a compact environment where neither of parties wants to disclose their private data. The secure two-party statistical analysis problem could be solvable in a way more efficient than the general circuit evaluation approach.

OPE is applied to SMC problems especially under privacy preserving data mining concept which is introduced in [12] by designing privacy preserving ID3 decision tree algorithm for purposes. A closer paper to our one is [11] where WAP is solved by two techniques. Former is by OPE and latter is by encryption techniques based on homomorphism yet both of them is used as a tool for k-means clustering for two
parties. Neural network applications are studied using OPE methods as well. Within the context of privacy-preserving data mining, [8] presented a private scalar product protocol based on standard cryptographic techniques and proved that it is secure.

3 Problem Definition, Cryptographic Tools and Criteria for Algorithm Design

In this section, the problem definition for privacy preserving two party and multiparty overall mean computation problems are defined mathematically whose algorithms are presented in sections 4 and 5, respectively. Moreover, the oblivious polynomial evaluation which is used as a cryptographic tool to construct the privacy requirements is briefly explained.

3.1 Privacy Preserving Two-Party Overall Mean Computation Problem

Suppose that Alice (party 1) has \( n \) samples \( A = \{ x_1, \ldots, x_n \} \) and Bob (party 2) has \( m-n \) samples \( B = \{ x_{n+1}, \ldots, x_m \} \). Each party wants to get the mean of their samples without revealing any private information. We are assuming that finding the mean of the union of samples from the two parties is more desirable than calculating the two samples individually.

The mean of Alice’s samples is \( \mu_a = (\sum_{i \in A} x_i) / n \) and Bob’s is \( \mu_b = (\sum_{i \in B} x_i) / (m-n) \) and overall mean of \( A \cup B \) is \( \mu = (\sum_{i \in A \cup B} x_i) / m \). But the computation should be done without computing the \( A \cup B \) so the means are weighted with respect to their cardinalities and joined together with multiplication and then divided by the union size. Hence,

\[
\mu = (\mu_a \cdot n + \mu_b \cdot (m-n)) / m
\]

yields us the overall mean where individual means are pre-computed and proportioned by their cardinalities. Remark the terms in the first product \( (\mu_a \cdot n) \) are only known by Alice and the terms in the second product \( (\mu_b \cdot n-m) \) are only known by Bob. Besides, the size of the union is known neither of them. So we get the following SMC problem.

**PROBLEM (Two Party OMP):** Alice has \( n \) samples represented by \( A = \{ x_1, \ldots, x_n \} \), and Bob has \( m-n \) samples represented by \( B = \{ x_{n+1}, \ldots, x_m \} \), where each \( x_i \) is a real number. Alice and Bob want to compute the overall mean of \( A \cup B \) without revealing any of their samples to the other principal which is according to the equation given in (1).
3.2 Privacy Preserving Multiparty Overall Mean Computation Problem

Assume that there exist $k$ parties denoted by $P = \{P_1, \ldots, P_k\}$ each having private data sets of cardinality $c$. For party $j$, the mean function can be calculated as $\mu_j = (\Sigma_{i \in j} i) / c$. Without loss of generality, the overall mean for all parties is

$$\mu = (\Sigma_i \mu_i c_i) / (\Sigma_i c_i), \quad 0 \leq \xi(k+1)$$

(2)

Our second and the main problem is given by the generalized version of the two party case where multiple parties conduct overall mean computation in the same environment without disclosing any private data represented by samples.

3.3 Oblivious Polynomial Evaluation

To design a secure protocol for computing a function $f(x, y)$ allows two parties, a receiver who knows $x$ and a sender who knows $y$, to jointly compute the value of $f(x, y)$ in a privacy preserving way. The fact that for every computable function $f(x, y)$ in polynomial time there exists such a (polynomially-computable) protocol is already achieved in the cryptographic research. In OPE problem, the input of the sender is a polynomial $P$ of degree $k$ over some field $F$. The receiver can get the value $P(x)$ for any element $x \in F$ without learning anything else about the polynomial $P$ and without revealing to the sender any information about $x$. If we define the input and output for the functionality of OPE, as a two party protocol run between a receiver and a sender over a field $F$, we get the big picture as:

- **Input**
  - Receiver: an input $x \in F$.
  - Sender: A polynomial $P$ defined over $F$.

- **Output**
  - Receiver: $P(x)$.
  - Sender: nothing.

There are various protocols to solve the OPE yet the protocol given by Naor and Pinkas [14] is preferred for OMP.

3.4 Criteria for Algorithm Design

To design a privacy preserving OMP algorithm, a number of assumptions should be made to overcome the deficiency of a trusted party. Semi-honest adversary model is chosen which is usually preferred while designing such an algorithm. The information disclosure is nothing but yielded by OPE protocol. No more breaches exist in the
specific computations regarding OMP. The complexity of the algorithm is in polynomial time since mean computation time complexity and OPE reduction $C_P$.

## 4 Privacy Preserving Two-Party OMP Protocol

In this section, privacy preserving two-party OMP protocol is designed. To develop such an algorithm, a functional should be taken as a target to place the terms in the OPE. Let $f$ be the functional for such a computation, its domain set must be two-dimensional vectors (i.e. mean and cardinality) for both parties and range set must be the same value (overall mean) as a two dimensional vector. Since the cardinalities are multiplied with individual means in the numerator and the total sum of samples exist in the denominator of (1), $f$ is constructed as:

$$f : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$f((x,m),(y,n)) = ((xm + yn)/(m + n), (xm + yn)/(m + n))$$

(3)

The straightforward solution is approximating $f$ by a circuit and using [9] for constructing the protocol for the privacy preserving case. However, it is well known that the cost for implementing such a circuit is so inefficient that a new solution should be developed for the specific case. We describe the protocol in a top-down fashion. Steps are as follows:

1. Define the private rational polynomial evaluation problem (RPE)
2. Produce a protocol for RPE using OPE.
3. Find a suitable case for RPE by placing polynomials and field elements for OMP. (Reduction from private-RPE $\rightarrow$ OMP)

**RPE.** For any finite field $F$, construct $f$ as

$$f : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$f((P,Q),(\alpha, \beta)) = (P(\alpha)/Q(\beta), P(\alpha)/Q(\beta))$$

(4)

where $P, Q \in F[x]$ (polynomials for party 1) and $\alpha, \beta \in F$ (field elements of party 2).

**RPE Protocol.** The protocol for RPE can be produced by OPE in the following way:

1. Party 1 “blinds” the polynomials $P$ and $Q$ by multiplying with a predetermined field element $\gamma \in F$.
2. Party 2 computes $\gamma P(\alpha)$ and $\gamma Q(\beta)$ by OPE twice.
3. Party 2 computes $P(\alpha)/Q(\beta)$, by computing $\gamma P(\alpha)/\gamma Q(\beta)$, and sends it to party 1.

The first two steps are adapted from [11] where the aim is to solve WAP which is relatively more trivial than OMP. The theorem which states RPE protocol is a privacy...
preserving one exists in the same work as well. In WAP, the target functional \( g \) is defined as:

\[
g : \mathbb{IR}^2 \times \mathbb{IR}^2 \rightarrow \mathbb{IR}^2 \\
g((x,m),(y,n)) = ((x+y)/(m+n), (x+y)/(m+n))
\]  

(5)

The crucial difference between WAP and OMP is that the former includes linear expressions in the numerator and denominator of its functional yet the latter has bilinear terms, i.e., \((x + y)\) versus \((xm + yn)\). Furthermore, the bilinear terms are coming from the individual parties so there is no obvious solution such as choosing dedicated polynomials for both numerator and denominator in the RPE problem. In the below, the construction of private-RPE \(\rightarrow\) OMP is stated by placing suitable polynomials and field elements:

**Reduction of private-RPE \(\rightarrow\) OMP.** Recall that party 1 has inputs \((x,m)\) and party 2 has inputs \((y,n)\). Since the reduction is from RPE, in the format of (4), party 1 needs to construct polynomials and party 2 needs to choose field elements. The polynomials for party 1 are:

\[
P(w) = w \cdot x \\
Q(w) = w \cdot m \\
R(w) = w \cdot x \\
S(w) = w \cdot m
\]  

(6)

Note that all polynomials are linear and coefficients are known by party 1. On the other hand, the field elements for party 2 are:

\[
\alpha = y \\
\beta = n \\
\gamma = -y \\
\psi = -n
\]  

(7)

The field elements \(\gamma\) and \(\psi\) are well-defined since every field is a group and inverse with respect to the addition exists for \(y\) and \(n\).

Let us define the \(T(w_1,w_2,w_3,w_4)\) as the linear combination of \(P\), \(Q\), \(R\) and \(S\) polynomials:

\[
T(w_1,w_2,w_3,w_4) = P(w_1)Q(w_2) + R(w_3) + S(w_4)
\]  

(8)

\(T\) is nothing but a dummy polynomial to handle the bilinear terms. Since multiplication and closed operations in a field \(T\) is also well-defined. If \((w_1,w_2,w_3,w_4) = (a, \beta, \psi, \gamma)\) variable replace is done then it yields:
The numerator of the desired functional \( f \) in (4) is constructed and the denominator is nothing but \( Q(t) \). Using OPE, the reduction is complete by choosing suitable field elements together with constructing the polynomial, \( T \) as a combination of linear polynomials.

Two lemmas are critical for the proof that \( f \) computes the overall mean privately. The former belongs to Canetti [2] and the latter belongs to Jha et al. [11]. Their proofs are not given here as they can be reached from original sources.

**Lemma 1. (Composition Theorem for passive adversary):** If \( g \) is privately reducible to \( f \) and there exists a protocol for computing \( f \) privately then there exists a protocol for computing \( g \) privately.

**Lemma 2. (Private RPE):** The protocol gives as RPE protocol privately computes RPE problem.

**Theorem 1. (Two-party OMP)** The protocol formed by \( f \) in (3) yields a privacy-preserving protocol for two-party OMP.

**Proof:** It is clear that OMP is privately reducible to RPE by choosing the numerator as \( T \) and the denominator as \( Q \) and there exists a protocol for private-RPE (Lemma 2) then by Lemma 1, given protocol is privately computes two-party OMP. It is trivial that reduction is polynomially-computable.

## 5 Privacy Preserving Multiparty OMP Protocol

The privacy preserving two-party OMP protocol can guide to extend the protocol for multi-party case. The protocol is presented step-by-step while explaining what is known to the both parties in crucial cases. Before starting to develop protocol, we need the following privacy-preserving protocol for cardinality summation.

**Lemma 3. (Private RPE):** The functional \( h \) given in (9) privately computes the sum of two samples whose cardinalities are \( m \) and \( n \) respectively two parties.

\[
h : \mathbb{IR}^2 \times \mathbb{IR}^2 \rightarrow \mathbb{IR}^2 \\
h ((x,m),(y,n)) = ((m + n), (m + n))
\]
Proof: The reduction to private-RPE completes the proof by joining lemma 2 with lemma 1. The unique polynomial for party 1 and the single field element of party 2 are as follows:

\[ Q(w) = w + m \]  \hspace{1cm} (10)

\[ \beta = n \]  \hspace{1cm} (11)

By computing \( Q(\beta) = m + n \), party 2 gets the desired result and shares with party 1.

**Multiparty OMP Protocol.** The protocol for multiparty OMP can be designed from two-party OMP in the following way:

1. Parties are ordered from 2 to \( k \) in a manner that consecutive parties are involved to two-party OMP computation. This can be done with a common share or coin-tossing into well protocol [9].

2. Between party \( j \) and \( j + 1 \), \( 0 < j < k \), the two-party OMP protocol works and

\[ \left( (\mu_j, c_j), (\mu_{j+1}, c_{j+1}) \right) \rightarrow \left( (\mu_j, c_j + \mu_{j+1}, c_{j+1}),(c_j + c_{j+1}) \right) \]

is computed.

3. Furthermore, party \( j \) and \( j + 1 \) privately compute their cardinality sum via the functional given in lemma 3. In other words, consecutive parties compute the partial mean and partial size of their samples.

\[ \left( (\mu_j, c_j), (\mu_{j+1}, c_{j+1}) \right) \rightarrow (c_j + c_{j+1}, \mu_j + \mu_{j+1}) \]

4. The mean and cardinality values are updated for party \( j + 1 \) with the new values calculated at the end of the protocol involved with the previously ordered party. (i.e. party \( j + 1 \) gets the partial mean and partial sum of samples up to her)

5. Apply the previous two steps for all consecutive parties. Total computation is linear in size and \( k \), for \( k \) parties.

6. At the end of the computation, the last party (i.e. party \( k \)) gets the overall mean together with total sample size with remaining parties.

The unimportant gap of the protocol is party \( j \) learns the size of the total previous samples yet it is not give the size of the each individual party. The only exception is for the first party, party 2 gets the size of its sample. This can be overcome by choosing the order in a circular round-robin fashion so the order of consecutive parties are preserved but only the first party changes. The probability to be the first party is \( 1/k \) which is Pareto-optimal for such a scheme.

**Theorem 2. (Multiparty OMP)** The protocol formed by \( f \) in (3) and \( h \) in (9) together yields a privacy-preserving protocol for multiparty OMP.

Proof: The part regarding \( f \) is done in the proof of the first theorem and the proof of lemma 3 shows the reduction for \( h \), thus by lemma 1, given protocol is privately computes multiparty OMP.
6 Conclusion and Future Work

In this paper, the application of the oblivious polynomial evaluation technique to secure multiparty computation problems is shown via presenting privacy preserving algorithm for multiparty overall mean computation using oblivious polynomial evaluation as a concrete example. To form a basis, the two-party case and cardinality summation algorithms are also given as components of the main protocol.

In our future directions, we have to adapt the private-OMP to a specific case, especially in data mining or machine learning. Principal Component Analysis [7] is chosen as a test bed which suits to the problem well since the mean computations are done before forming the covariance matrix. The controlled experiments can guide us to justify that the protocol works without trouble.

Moreover, an algorithm that uses one of the homomorphic encryption schemes (HES) can be designed for the same problem. The scheme developed by Paillier [15] can be considered to be the alternatives for such a mechanism. A testing environment for the comparison of the performance analysis between OPE and HES determines the efficient algorithm under given security assumptions.

References

16. Pinkas B.: Cryptographic Techniques for Privacy-Preserving Data Mining SIGKDD Explorations, the newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, (2003)