IADIS International Journal on Computer Science and Information Systems
Vol. 11, No. 1, pp. 32-48
ISSN: 1646-3692

SECURE HEALTH STATISTICAL ANALYSIS METHODS

Saeed Samet. *eHealth Research Unit, Faculty of Medicine, Memorial University St. John's, NL, Canada*

Ahoora Sadeghi Boroujerdi. *Department of Computer Science, Faculty of Science, Memorial University, St. John's, NL, Canada*

Shabnam Asghari. *Primary Healthcare Research Unit, Faculty of Medicine, Memorial University St. John's, NL, Canada*

ABSTRACT

Health informatics, using new information technology, provides a fruitful set of data resource and knowledge that is very useful for secondary data users and researchers in various health systems and applications. Privacy acts, on the other hand, prevent direct access to this information without patient's consent. Various solutions have been proposed such as data anonymization and de-identification, on-site analysis, and limited remote access, to preserve the data owner's privacy. Each of those approaches has different drawbacks and limitations. For instance, data de-identification will reduce data utility because of low precision of the final released data, and also it has a risk of data re-identification using available public data and background knowledge. On-site analysis has physical limitations, such as lack of data centers in every geographic area, and time-consuming procedures, such as background checks. Remote access increases security risks, and when data has to be pulled from multiple data resources, it requires patient consent for data disclosure. In this paper, we propose a set of privacy-preserving methods and techniques for some popular health statistical analysis methods. Using this set of secure protocols health researchers and other data users are able to issue their requests as some queries, and receive only the results of their queries from the data owners, while each data custodian can keep their sensitive data private. Proposed methods have been tested using sample data to illustrate the performance of the results in terms of computational and communication complexities. Security proof of the proposed protocols has also been provided as a proof of concept.

KEYWORDS

Privacy-Preserving; Secure Multiparty Computation; Health Informatics; Homomorphic Encryption; Health Statistics.

1. INTRODUCTION

With the very fast growing of electronic health information collected in various health facilities, there are increasing demands for disclosing this information for secondary purposes, such as health services research and public health. One approach that is gaining interest is to provide the data recipient remote access to the data over the Internet. In addition to the increased security risks from providing remote access to data, if the data recipient can access the data, even remotely, then this is a disclosure, which requires patient consent. Furthermore, remote access does not solve the problem when data needs to be pooled from multiple sites and analyzed together. There are number of general approaches to de-identify data. First, it is possible to generalize and suppress the individual level data and disclose it. An ideal approach is to disclose encrypted data to the end-user, and allow the end-users to perform their statistical analysis on the encrypted values. This method of secure computation ensures that the data recipient does not get any viewable patient data, but still allows them to run their own analyses and perform their own diagnostics on the resultant models. However, this approach needs a fully homomorphic cryptosystem, such that data user can perform every operation on encrypted values. This approach is still under investigation by many researchers, and no practical solution has been proposed yet. A practical and secure solution has been introduced in this work, in which data owners will securely and jointly perform a privacy- preserving protocol by only exchanging encrypted values between themselves, and at the end, each of them will send their portion of the final results to the data user, who then combines the received shares to construct the final results of her query.

The advantage of secure computation over traditional de-identification methods, such as generalization and suppression, is that the risk of re-identification is zero, and the results will not suffer from the lower precision introduced by generalization and suppression. However, secure computation can be slower than running the same analyses on de-identified data. Furthermore, many statistical analysis methods on encrypted data have not been developed yet. In this paper secure computation techniques for common data analysis methods, such as Mean, Variance, Skewness, and Chi-square tests are proposed. In our scenario, there are two types of parties involved in the protocol, Data Owner, and Data User (or Researcher). Each data user has a subset of records from the whole dataset, such that no other party has access to that portion of plain data. Data user sends her request to the data owners and receives partial information from each of them. By performing local operations on received information, data user will calculate the final results of her query.

A current approaches is that some data custodians, such as Statistics Canada, provide researchers access to potentially identifiable information at special Research Data Centres (RDCs). To use these secure facilities however the researchers have to take an oath of confidentiality and undergo a very time consuming process due to approval and background checking. In addition to the considerable start-up delay, RDC type facilities require the researcher to be physically present at the RDC to conduct any analysis, and the provided data is not up-to-date and is very limited. These centres do not let access to Internet and/or electronic transfer of your output, and the output results should be reviewed and approved by the centres and the results might be partially omitted if there is a risk of confidentiality breach. In addition, there are only few such centres for access to data across the countries. Also, some custodians disclose de-identified data to the researchers, which reduces the precision of the data and results in the suppression of data cells. Our approach, on the other hand, allows the

researcher to conduct the analysis from anywhere and get started relatively quickly, the output is developed in direct collaboration between researcher and data owner, it removes the risk of results inaccuracy, increases the data sharing speed in contrast of he current process of de-identification and data extractions procedures that take a long time, and makes these data more suitable for timely decision making. Also, in traditional ways sometimes the data owners do not provide the data in the requested quantity, quality and format due to security reasons.

The main objective of this paper is to propose a new methodology for secure statistical analysis methods, which allows the researchers to analyze the original data without any loss of precision or suppression, while data privacy is preserved. In statistical analysis methods, we are dealing with different computations with various mathematical operations such as addition, multiplications, exponentiation and natural logarithms. To have secure computations for this set of operations we need secure and efficient privacy homomorphism techniques to apply on those methods. However, most of the homomorphic encryption methods only support one operation, addition or multiplication, and the existing cryptosystems that are fully homomorphic such as (Ferrer 96-1,2) have various security vulnerabilities, and are not currently practical for real-world privacy-preserving protocols.

The rest of this paper is structured as follows: In Section 2 background and related work are reviewed. Secure building blocks required for the main protocol are introduced in Section 3. Configuration of the parties involved in the scenario, and the secure protocol for data analysis techniques are proposed in Section 4 along with security and complexity analysis. Experimental results will be shown in Section 5, followed by the conclusions and future work in Section 7.

2. BACKGROUND

Since 2000, when two seminal papers (Lindell and Pinkas 2000, Agrawal and Srikant 2000), both entitled Privacy-Preserving Data Mining, were published, research in the field of secure computation has dramatically increased and many protocols and algorithms have been proposed for different standard data mining, machine learning, and data analysis techniques (Vaidya, Clifton and Zhu 2006, Aggarwal and Yu 2008). Some of those protocols use randomization and perturbation approach to preserve the privacy of the data owner by adding noise to the original data. However, this approach suffers from inaccuracy of the final results and also lack of strong security. Other privacy-preserving computation protocols utilize Secure Multi-party Computation (SMC) to generate the final results in a secure way among multiple parties.

Cryptographic and other tools, such as oblivious transfer protocol, are often used among two or more parties to jointly and securely compute one or more functions using their own private inputs. By using this approach, the final result is the same as that in the corresponding non-secure algorithm, and thus the main trade-off is between security and efficiency. Because of the complex computations in data analysis techniques, secure building blocks are proposed and utilized inside the main protocols in different steps of their algorithms. Examples are secure sum (Clifton et al. 2002), secure comparison (Yao 1982), and secure multi-party multiplication and factorial (Samet and Miri 2009).

Another work on secure multi-party computation is done by Karr et al. (Karr 2009, Karr et al. 2007). Although a modified version of the secure sum (Clifton et al. 2002), which has some security vulnerabilities, has been presented, the new method has also some problem, such as using a server, which is a bottleneck for the whole system. Other problems also exist in the statistical analysis method, such as computing inverse of matrices that are distributed among the parties, which are mentioned in (Karr 2009) by the primary author of (Karr et al. 2007).

3. SECURE BUILDING BLOCKS

In this section, we describe secure sub-protocols used inside our main protocol as building blocks. Data mining algorithms and statistical analysis methods are usually complex, and contain more than one simple operation. Therefore, to add security features to preserve the privacy of the original data, secure building blocks are utilized inside those algorithms and operations. One of the very popular encryption techniques used in privacy-preserving methods is homomorphic encryption. Some existing cryptosystems, such as Paillier (Paillier 1999), RSA (Rivest, Shamir and Adleman 1978), and Elgamal (Elgamal 1985) support a homomorphic encryption with an acceptable security proof. However, in each encryption system, one type of operation, addition or multiplication, is supported. Therefore, in many statistical analyses they could not be utilized. We use Paillier encryption method, which is an additive homomorphic encryption system along with secure multiplication (Samet and Miri 2009), to overcome the lack of multiplicative homomorphism, in our proposed protocol. In the Paillier cryptosystem addition on the plaintexts will be mapped to the multiplication on the corresponding ciphertexts, i.e. for any two plaintext messages m_1 and m_2 , and their encryptions, $Enc(m_1)$ and $Enc(m_2)$, the following equation is satisfied:

$$\mathrm{Dec}(\mathrm{Enc}(\mathrm{m}_1) * \mathrm{Enc}(\mathrm{m}_2)) = \mathrm{m}_1 + \mathrm{m}_2$$

In the above equation, Enc and Dec indicate encryption and decryption, respectively. Another feature of this cryptosystem is:

$$Dec(Enc(m_1)^{m_2}) = m_1 * m_2$$

One issue that has to be considered in the usage of the cryptosystems like Paillier is that we are dealing with integer numbers while the real-world data has usually no restriction, and could be any real numbers. To overcome this problem the input data will be scaled and rounded before the encryption, and the final results could be rescaled to its correct decimal point after the decryption.

We also utilize secure building block Secure Multiparty Addition (SMA) (Samet and Miri 2009). For instance, using SMA parties $P_1, P_2, \dots, P_n$, have their own private input shares, $x_1, x_2, \dots, x_n$, and will reach to their private output shares, $y_1, y_2, \dots, y_n$, such that:

$$\sum_{i=1}^{n} x_i = \prod_{i=1}^{n} y_i$$

This building block uses Paillier cryptosystem to preserve the privacy of the parties' inputs and outputs. Note that all the mathematical computations are modular. For instance, if the operations are done in mod n=35, then 15+24 = 9*16 because 9*16 = 144 = 4 mod 35. However, in real applications, mod number is very large.

4. PRIVACY-PRESERVING STATISTICAL ANALYSIS METHODS

In this section we will show the protocols for the popular statistical analysis methods. Without loss of generality, we assume that there are two data owners involved in each protocol. The multi-party scenario is the same as that in two-party case, with some extra computation and data communication among the parties. In addition, data users are securely communicating with the data owners through a portal, by which they can send their queries and receive the results back from the data owners. Metadata of the datasets, such as attribute names are assumed to be public or exchanged among the data users and owners as a pre-processing task, as it is illustrated in Figure 1.

Initial configuration for the rest of this section is as follows, and illustrated in Figure 1:

- Each data owner, D_i, has n_i records (data rows) from the whole n records of the dataset, i.e. $n = \sum_{i=1}^{k} n_i$, in which k is the number of data owners.

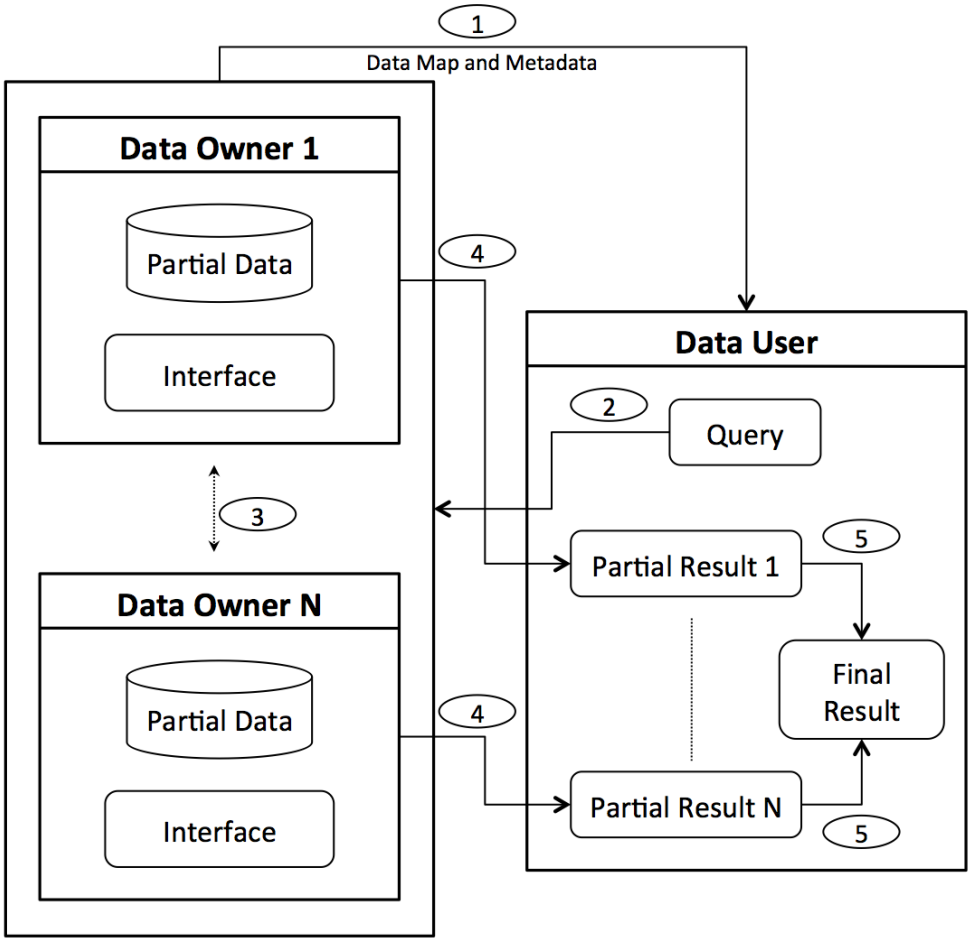

Figure 1. Data and process flow of the protocol

- One data owner will establish the encryption keys for secure data exchange among them. Without loss of generality, we set D_1 for this task. Therefore, every data owner D_i can encrypt data, but only D_1 is able to decrypt any encrypted information received from the other data owners.

4.1 Mean

In this protocol, data user will send the name of the attribute (data column), say A, to $D_i's$ to receive the mean value of that attribute from the whole dataset, that is securely shared between the data owners, D_1 and D_2. Note that $A = \langle a_1, a_2, \dots, a_n \rangle = \langle a_{1,1}, a_{1,2}, \dots, a_{1,n_1}, a_{2,1}, a_{2,2}, \dots, a_{2,n_2} \rangle$, such that $a_{i,n_i} \in D_i$ and $n = n_1 + n_2$.

$$\mu = \frac{\sum_{i=1}^{n} a_i}{n} = \frac{\sum_{i=1}^{n_1} a_{1,i} + \sum_{i=1}^{n_2} a_{2,i}}{n_1 + n_2}$$

1. Each data owner D_i will compute the summation of attribute's values from her own records, such that

$$A_i = \sum_{j=1}^{n_i} a_{i,j}$$

2. D_1 will encrypt A_1 and send $Enc(A_1)$ to D_2.

3. D_2 will randomly generate her output share, B_2 , encrypt A_2 , and send the following information to D_1:

$$(Enc(A_1) * Enc(A_2))^{B_2^{-1}}$$

4. D_1 will decrypt the value received from D_2 , and set it as her own private output, B_1. Note that:

$$\begin{aligned} B_1 &= Dec((Enc(A_1) * Enc(A_2))^{B_2^{-1}}) \\ &= Dec((Enc(A_1 + A_2))^{B_2^{-1}}) \\ &= Dec(Enc(B_2^{-1} * (A_1 + A_2))) \\ &= B_2^{-1} * (A_1 + A_2) \end{aligned}$$

Last equation above satisfies $A_1 + A_2 = B_1 * B_2$.

Steps 2 to 4 are actually the main stages of secure two-party addition (S2A).

5. Data owners will perform S2A for their two private values n_1 and n_2 as well, such that:

$$n_1 + n_2 = r_1 * r_2$$

6. Each data owner D_i will securely send $\langle B_i, r_i \rangle$ to the data user.

7. Data user will compute $= \frac{B_1 * B_2}{r_1 * r_2}$, which is the mean value for the attribute A.

Note that none of the values $\langle B_i, n_i \rangle$ and their combinations will reveal individual private information of the data owners, which preserves their data privacy.

4.2 Variance

Suppose data user wants to receive the variance of the attribute $A = \langle a_{1,1}, a_{1,2}, \dots, a_{1,n_1}, a_{2,1}, a_{2,2}, \dots, a_{2,n_2} \rangle$.

$$v = \frac{\sum_{i=1}^{n} (a_i - \mu)^2}{n} = \frac{\sum_{i=1}^{n_1} (a_{1,i} - \mu)^2 + \sum_{i=1}^{n_2} (a_{2,i} - \mu)^2}{n_1 + n_2}$$

1. Each data owner D_i will perform the following local computations on the attribute's values from her data records:

$$A_i = \sum_{j=1}^{n_i} a_{i,j} \quad , \quad S_i = \sum_{j=1}^{n_i} a_{i,j}^2$$

2. Data owners will perform S2A for their two private values A_i , S_i , and n_i , such that:

$$A_1 + A_2 = B_1 * B_2$$
$$S_1 + S_2 = T_1 * T_2$$
$$n_1 + n_2 = r_1 * r_2$$

3. Each data owner D_i will securely send $\langle B_i, T_i, r_i \rangle$ to the data user.

4. Data user will compute $= \frac{T_1 * T_2}{r_1 * r_2} - \frac{(B_1 * B_2)^2}{(r_1 * r_2)^2}$, which is the variance of the attribute A.

4.3 Skewness

To compute skewness of an attribute $A = \langle a_{1,1}, a_{1,2}, \dots, a_{1,n_1}, a_{2,1}, a_{2,2}, \dots, a_{2,n_2} \rangle$, based on the following equation for this statistical method, data user will send three separate queries, to receive partial results from each data owner.

$$\gamma = \frac{\sum_{i=1}^{n}(a_i - \mu)^3}{n\sigma^3} = \frac{1}{\sigma^3}\frac{\sum_{i=1}^{n}(a_i - \mu)^3}{n} = \frac{1}{\sigma^3}\frac{\sum_{i=1}^{n_1}(a_{1,i} - \mu)^3 + \sum_{i=1}^{n_2}(a_{2,i} - \mu)^3}{n_1 + n_2}$$

The first query is for receiving the partial results of the mean, μ, of the attribute, the second query is for receiving the partial results of the standard deviation, σ, of the attribute, and the third query is for the rest of the equation.

1. Means and standard deviation of the attributes can be calculated using the two previous protocols.

2. Each data owner D_i will perform the following local computations on the attribute's values from her data records:

$$A_i = \sum_{j=1}^{n_i} a_{i,j}^2 \quad , \quad S_i = \sum_{j=1}^{n_i} a_{i,j}^3$$

3. Data owners will perform S2A for their two private values A_i , S_i , and n_i , such that:

$$A_1 + A_2 = B_1 * B_2$$
$$S_1 + S_2 = T_1 * T_2$$
$$n_1 + n_2 = r_1 * r_2$$

4. Each data owner D_i will securely send $\langle B_i, T_i, r_i \rangle$ to the data user.

5. Data user will compute $= \frac{1}{\sigma^3}\left(\frac{T_1 * T_2}{r_1 * r_2} + 2\mu^3 - \frac{3\mu(B_1 * B_2)^2}{r_1 * r_2}\right)$, which is the variance of the attribute A.

4.4 Correlation

To compute the correlation between two attributes A and , we have to compute:

$$\rho = \frac{(\sum_{i=1}^{n} a_i * b_i) - n * \mu_A * \mu_B}{n * \sigma_A * \sigma_B}$$

μ_A and μ_B are the mean values, and σ_A and σ_B are the standard deviations of the attributes A and B, respectively, and

$$A = \langle a_1, a_2, \dots, a_n \rangle = \langle a_{1,1}, a_{1,2}, \dots, a_{1,n_1}, a_{2,1}, a_{2,2}, \dots, a_{2,n_2} \rangle$$

$$B = \langle b_1, b_2, \dots, b_n \rangle = \langle b_{1,1}, b_{1,2}, \dots, b_{1,n_1}, b_{2,1}, b_{2,2}, \dots, b_{2,n_2} \rangle$$

1. Means and standard deviations of the attributes can be calculated using the previous corresponding protocols.

2. Each data owner D_i will perform the following local computation on the attribute's values from her data records:

$$S_i = \sum_{j=1}^{n_i} a_{i,j} * b_{i,j}$$

3. Data owners will perform S2A for their two private values S_i , such that:

$$S_1 + S_2 = T_1 * T_2$$

$$n_1 + n_2 = r_1 * r_2$$

4. Each data owner D_i will securely send $\langle T_i, r_i \rangle$ to the data user.

5. Data user computes $= \frac{T_1 * T_2 - r_1 * r_2 * \mu_A * \mu_B}{r_1 * r_2 * \sigma_A * \sigma_B}$, which is the correlation of the selected attributes.

4.5 Chi-Square Test

To compute chi-square test, we should first compute counts for the four possible combinations of specified categorical variables. Therefore, a secure count protocol is needed to compute the counts for each cell of the contingency table.

4.5.1 Count

Suppose data user wants to know the number of records such that the value of the attribute A is m.

1. Each data owner D_i will locally compute the count of the records in which the attribute A has the value m. Suppose m_i is this count value for D_i.

2. Data owners will perform S2A for their two private values m_i, such that:

$$m_1 + m_2 = c_1 * c_2$$

3. Each data owner D_i will securely send c_i to the data user.

4. Data user will compute $c = c_1 * c_2$, which is the count of the rows with the value m for the attribute A.

Now suppose C_i is the count for cell i in the contingency table, Table 1.

Table 1. Contingency table.

	Yes	No
Yes	C_1	C_2
No	C_3	C_4

C_1, C_2, C_3, and C_4 are the counts for the possible combinations. Therefore, the data user will calculate chi-square of the above contingency table as follows:

$$\chi^2 = \frac{(C_1 * C_4 - C_2 * C_3)^2 * (C_1 * C_2 * C_3 * C_4)}{(C_1 + C_2) * (C_3 + C_4) * (C_2 + C_4) * (C_1 + C_3)}$$

Odds Ratio, if it is defined in terms of the joint probability distribution of two binary random variables, could also be calculated using the above chi-squared method.

Using other secure building blocks, such as secure dot product (Goethals 2004) and secure matrix inverse addition (Han, Ng, Yu 2008), logistic regression can also computed in a privacy-preserving manner.

5. SECURITY AND COMPLEXITY ANALYSES

For the security analysis of the proposed protocols, we assume that the data owners involved follow all the steps of the protocol by properly performing the operations and exchanging correct data with each other. However, they may use the intermediate results to reach private information of each other. We use simulation paradigm (Goldreich 2004, Goldwasser, Micali, Rackoff 1989) along with the composition theorem to prove the security of the protocols. Using simulation paradigm, a protocol is considered secure if all the received data by a data owner can also be obtained by her inputs and outputs. Thus, for each data owner D we have to find a simulator S such that its output is computationally indistinguishable (Goldreich 2004) from that party's view using the secure protocol.

Composition theorem helps the security proof of the complex protocols, in which each protocol is composed of some sub-protocols such that the inputs of one sub-protocol are the outputs of the previous one. As the data exchange between the data owners are similar in the proposed protocols, we only analyse security of one of the protocols, Mean. We also show the security analysis of the secure two-party addition that we have used as a secure building block inside the protocols. Note that all the data communications among the data owners and users

in our proposed protocols will be performed under secure channels with proper authentication and digital signature techniques.

5.1 Mean

The first step of computing the mean value of an attribute, there is no data exchange between the data owners, and therefore no security proof is needed.

In the second step, first data owner, D_1, sends an encrypted data to the second one, D_2, who has no access to the encryption private key and therefore is not able to decrypt the received encrypted value.

In the third step, D_2 computes her private shares, multiplies by the received value from D_1, and sends to D_1. Thus the simulator for this party would be:

Input: An encrypted value $Enc(A_1)$

Process:

- Generating a random number B_2
- Computing $\left(Enc(A_1) * Enc(A_2)\right)^{B_2^{-1}}$

Output: $\left(Enc(A_1) * Enc(A_2)\right)^{B_2^{-1}}$ (To D_1)

D_2 has no information about $D_1's$ private key, and thus will not know anything about $Enc(A_1)$. Also, D_1 has no information about $D_2's$ random number, and even after decryption of the received value from D_2, is not able to reveal $D_2's$ private information.

5.2 Secure Two-Party Addition

Suppose, there are two parties, D_1 and D_2, each of which has a private input, x_1 and x_2 respectively. We denote the protocol of secure two-party addition by φ and the desired functionality of φ by $f(x_1, x_2)$, such that:

$$f: X \times X \rightarrow Y \times Y.$$

Furthermore, we show the first and second elements of $f(x_1, x_2)$ by $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$, which are the private outputs of D_1 and D_2, respectively. Also, the view of D_1 (resp. D_2) during the execution of φ on (x_1, x_2) is denoted by $\text{VIEW}_1^{\varphi}(x_1, x_2)$ (resp. $\text{VIEW}_2^{\varphi}(x_1, x_2)$). Therefore, we have the following two equations:

$$VIEW_1^{\varphi}(x_1, x_2) = \{x_1, y_1, E_1(y_1, e_1)\}$$

$$VIEW_2^{\varphi}(x_1, x_2) = \{x_2, y_2, E_1(x_1, e_1)\}$$

Note, that as the D_2's point of view, $E_1(x_1, e_1)$ is just an encrypted value produced by and received from D_1 to that party. We say that protocol φ privately computes f, if two polynomial-time algorithms V_1 and V_2 exist, such that:

$$\{V_1(x_1, f_1(x_1, x_2))\} \stackrel{c}{\equiv} VIEW_1^{\varphi}(x_1, x_2)$$

$$\{V_2(x_2, f_2(x_1, x_2))\} \stackrel{c}{\equiv} VIEW_2^{\varphi}(x_1, x_2)$$

In each of the above equations, symbol $\stackrel{c}{\equiv}$ means that the two ensembles in both sides of the equivalence symbol are computationally indistinguishable.

Proof: For V_1 , suppose D_1 is corrupted by an adversary. Thus, the adversary knows all the items in the set of D_1's view in the protocol φ, i.e. the set of $\{\mathrm{x}_1, \mathrm{y}_1, \mathrm{E}_1(y_1, e_1)\}$. Obviously, simulator V_1 would be trivially as follows:

Input: $(\mathrm{x}_1, \mathrm{y}_1)$

Process: Computing $\mathrm{E}_1(y_1, e_1)$

Output: $\{\mathrm{x}_1, \mathrm{y}_1, \mathrm{E}_1(y_1, e_1)\}$

For the simulator V_2, suppose D_2 is corrupted by an adversary. D_2's view in φ is the set of $\{\mathrm{x}_2, \mathrm{y}_2, \mathrm{E}_1(x_1, e_1)\}$ in which, as it is previously indicated, $\mathrm{E}_1(x_1, e_1)$ is considered an unknown value received by D_2. Therefore, the simulator V_2 has to generate a random number and send it to D_2 each time D_2 needs to get a message from D_1. Thus, V_2 could be as follows:

Input: $(\mathrm{x}_2, \mathrm{y}_2)$

Process: Generating a random number r

Output: $(\mathrm{x}_2, \mathrm{y}_2, r)$

As it is shown above we can conclude that the incoming messages of D_1 and D_2 in the protocol φ, and incoming messages of D_1 and D_2 from V_1 and V_2 , respectively, are indistinguishable.

Similar proof could be shown for multi-party case and other protocols in this paper, by utilizing composition theorem (Goldreich 2004, Canetti 2000).

6. EXPERIMENTAL RESULTS

To measure the performance of the methods presented in this work in the real-world applications, we have implemented the cryptosystem used in this paper along with the secure building blocks utilized inside the main protocol, by applying on a synthesized dataset with 10,000 rows of information. Java programming language has been selected as a platform-independent language for developing the system and MySQL has been used for database. To investigate the performance of the proposed protocols, we first convert all the data to "BigInteger" for calculation purposes, as a pre-processing stage, and store the converted data in another database. This stage will only have to be executed once or when the stored data changes. Then, we implement the building block; secure addition, which is utilized inside the privacy-preserving methods proposed in this paper. The hardware and software specifications for the experiment are as follows:

- Windows 7 Professional
- Intel® Core™ i5-4570 3.20GHz
- 8 GB DDR3 RAM

Table 2 shows the performance of pre-processing stage for different data types and mixed data types. By mixed data types, we mean converting all data types at the same time.

Table 2. Performance results for the pre-processing stage.

Data Type	Time (in Seconds)
Integer	2.168
Date Time	2.949
String	2.767
Double	2.889
Mixed (for 10 different columns)	36.307

Table 3 shows the performance of the encryption, decryption and secure addition. The encryption key length in this experiment is 1024 bits. Although this key length is secure enough for most of the health datasets, it could be larger, e.g. 2048 bits, to achieve higher level of security, if needed.

Table 3. Performance results for the cryptosystem and secure addition.

Algorithm	Time (in Seconds)
Encryption	0.00014
Decryption	0.00980
Secure Addition	0.08726

Table 4 illustrates the overall time for each protocol when the number of records is 10,000 (fetching the data from database is also included in the calculated time).

Table 4. Performance results for the protocols (in Seconds) including fetching data from database.

Algorithm	Overall Time (in Seconds)
Count	0.284
Mean	0.468
Variance	0.613
Correlation	1.366
Chi-square	0.748
Skewness	0.956

Table 5 illustrates the overall time for each protocol when the number of records is 10,000 if the data is stored in memory.

Table 5. Performance results for the protocols (in Seconds) when data is stored in memory.

Algorithm	Overall Time (in Seconds)
Count	0.087
Mean	0.191
Variance	0.327
Correlation	0.854
Chi-square	0.489
Skewness	0.575

We have also implemented a portal used by a given researcher as the data user to send their queries to the data owners and receive the final results back from them without knowing the individual records. Figures 2 to 5 illustrate generating and submitting queries and receiving their results. Using two lists, the researcher can choose attribute(s), as well as the

statistical analysis method, and submit the request to the data owners. Partial query results will be securely computed and sent back to the portal for the final computation and showing the query result to the researcher.

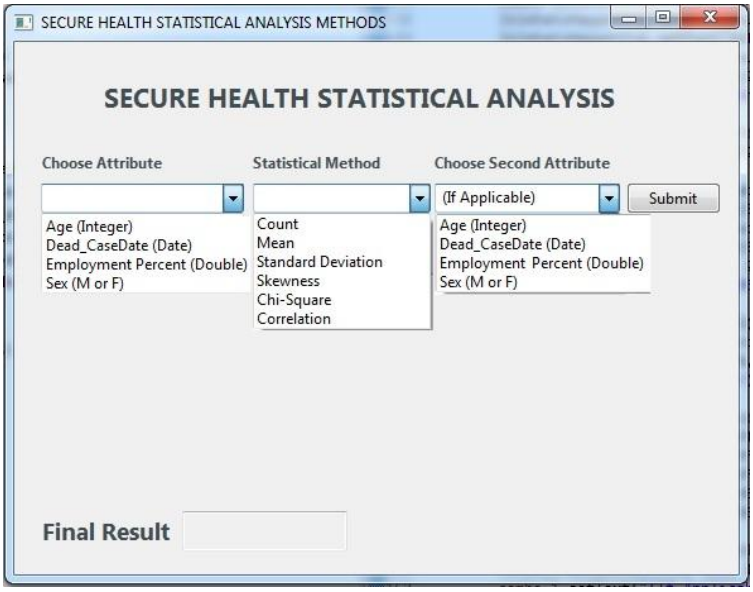

Figure 2. Researcher (data user) selects attribute(s) and the statistical analysis method and submits the query to the data owners.

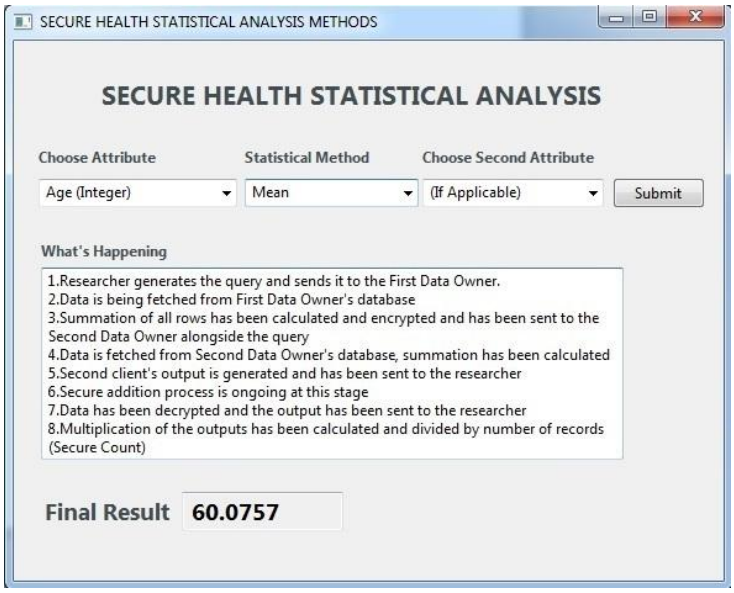

Figure 3. Data process to compute Mean value of the attribute Age is done on the data owners' side and the final result will be shown on the researcher's portal.

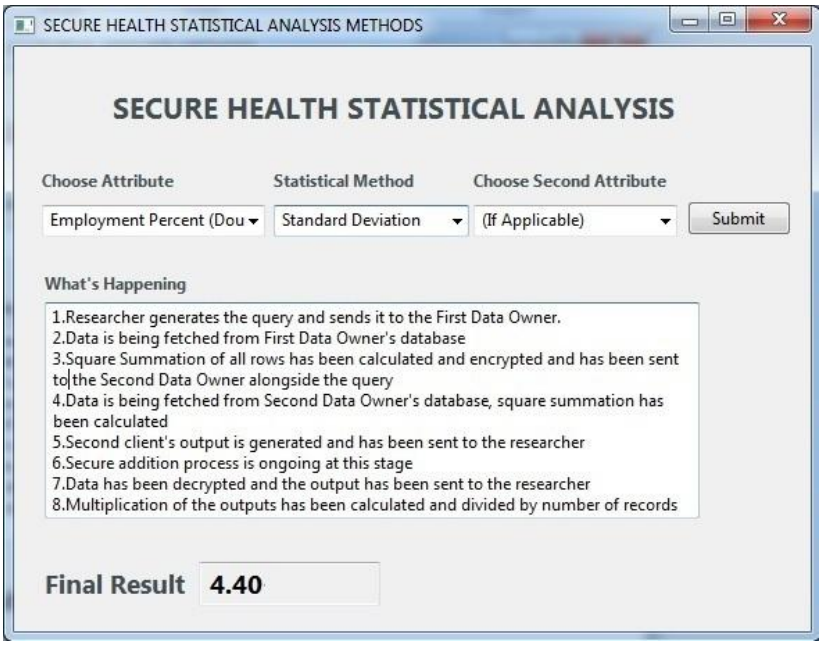

Figure 4. Data process to compute Standard Deviation value of the attribute Employment_Percent is done on the data owners' side and the final result will be shown on the researcher's portal.

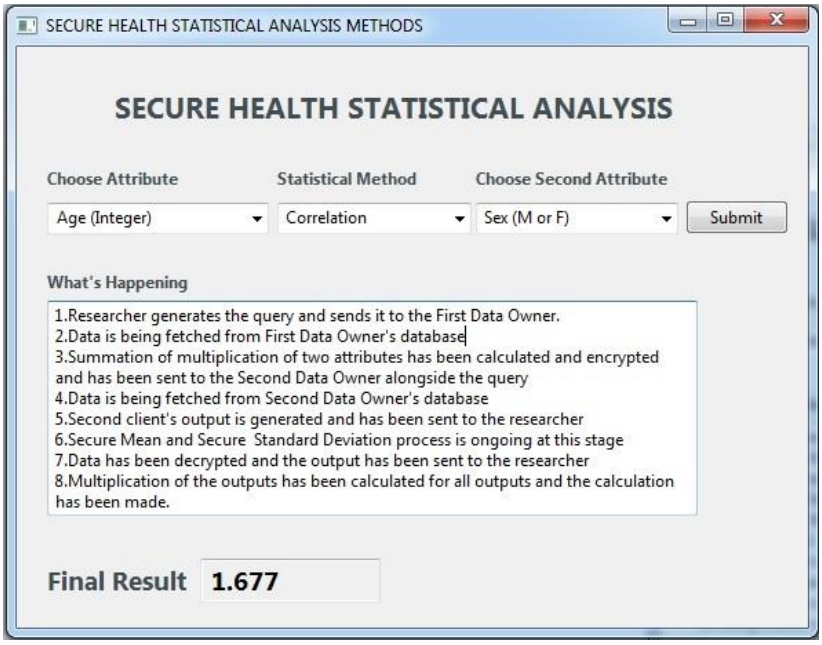

Figure 5. Data process to compute Correlation value of the attributes Age and Sex is done on the data owners' side and the final result will be shown on the researcher's portal.

7. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed privacy-preserving protocols for popular statistical analysis methods used in various applications, especially in health records, in which we need to maintain the privacy of the sensitive patient's information. During the execution of the protocol, computations and secure data communications are done by two or more data owners on the private distributed data, and only the final results will be decrypted and sent back to the data user. We are currently experimenting the proposed protocols on a Medico-administrative dataset from Newfoundland and Labrador Center for Health Information to illustrate their applicability.

As the future work, we are currently extending our protocols to cover other important statistical methods, which are extensively used in health research and other fields of science. Integrating the platform with statistical analysis software is another approach we are working on. Also, protocols should be expanded to cover another scenario, in which the collaboration of the multiple data custodians is done heterogeneously, i.e. each data custodian has the information for a subset of attributes from all data records. Another possible scenario would be distributing each single value of an attribute from each record among multiple parties, such that no party owns the whole value of the attribute. However, they can jointly perform the protocol on their shares to provide the final query results to the data user.

ACKNOWLEDGEMENT

This work has been partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Research & Development Corporation of Newfoundland and Labrador (RDC).

REFERENCES

Agrawal, R. & R. Srikant 2000, *Privacy-Preserving Data Mining*. In The ACM Special Interest Group on Management of Data Conference (SIGMOD), Dallas, TX, USA, pp. 439-450.

Aggarwal, C.C. & P.S. Yu 2008, *Privacy-Preserving Data Mining: Models and Algorithms*. Springer, vol. 513.

Canetti, R. 2000. *Security and Composition of Multiparty Cryptographic Protocols*. Journal of Cryptology. 13(1): pp. 143-202.

Clifton, C., et al. 2002, *Tools for Privacy-Preserving Distributed Data Mining*. SIGKDD Explorations, Newsletter, vol. 4, no. 2, pp. 28-34.

Domingo-Ferrer, J. 2002, *Inference Control in Statistical Databases: From Theory to Practice*. Lecture Notes in Computer Science, vol. 2316.

ElGamal, T. 1985, *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*. IEEE Transactions on Information Theory, vol. IT-31, no. 4, pp. 469-472.

Ferrer, J.D.I. 1996-1, *A new privacy homomorphism and applications*. Information Processing Letters, vol. 60, no. 5, pp. 277-282.

Ferrer, J.D.I. 1996-2, *Privacy homomorphisms for statistical confidentiality*. Questiio: Quaderns d'Estadistica, Sistemes, Informatica i Investigació Operativa, vol. 20, no. 3, pp. 505-525.

Goethals, B., et al. 2004, *On Private Scalar Product Computation for Privacy-Preserving Data Mining*. In The 7th International Conference on Information Security and Cryptology (ICISC), Seoul, Korea.

Goldreich, O. 2004. *Foundations of Cryptography: Volume 2, Basic Applications*. New York, NY, USA: Cambridge University Press.

Goldwasser, S., S. Micali, and C. Rackoff, 1989. *The Knowledge Complexity of Interactive Proof Systems*. SIAM Journal on Computing,. 18(1): p. 186-208.

Han, S., W.K. Ng, and P.S. Yu 2008, *Privacy-preserving linear fisher discriminant analysis*. In The 12th Pacific-Asia conference on Advances in knowledge discovery and data mining, Springer-Verlag: Osaka, Japan, pp. 136 -147.

Karr, A.F. 2009, *Secure Statistical Analysis of Distributed Databases, Emphasizing What We Don't Know*. Journal of Privacy and Confidentiality, vol. 1, no. 2, pp. 197-211.

Karr, A.F., et al. 2007, *Secure, privacy-preserving analysis of distributed databases*. Technometrics, vol. 49, no. 3, pp. 335-345.

Lindell, Y. & B. Pinkas 2000, *Privacy Preserving Data Mining*. In The 20th Annual International Cryptology Conference (CRYPTO), Santa Barbara, CA, USA, pp. 36-54.

Paillier, P. 1999, *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*. In the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT), Prague, Czech Republic, pp. 223-238.

Rivest, R., A. Shamir, & L. Adleman 1978, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of the ACM, vol. 21, no. 2, pp. 120-126.

Samet, S. & A. Miri, *Privacy-Preserving Bayesian Network for Horizontally Partitioned Data*. In The 2009 IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT2009), Vancouver, Canada, pp. 9-16.

Vaidya, J., C.W. Clifton, & M. Zhu 2006, *Privacy Preserving Data Mining*, Springer, vol. 120.

Yao, A.C. 1982, *Protocols for Secure Computations*. In The 23rd Annual Symposium on Foundations of Computer Science (SFCS), IEEE Computer Society, Washington, DC, USA, pp. 160-164.