Privacy-Preserving Linear Fisher Discriminant Analysis

Shuguo Han and Wee Keong Ng

School of Computer Engineering, Nanyang Technological University, Singapore
{hans0004,awkng}@ntu.edu.sg

Abstract. Privacy-preserving data mining enables two or more parties to collaboratively perform data mining while preserving the data privacy of the participating parties. So far, various data mining and machine learning algorithms have been enhanced to incorporate privacy preservation. In this paper, we propose privacy-preserving solutions for Fisher Discriminant Analysis (FDA) over horizontally and vertically partitioned data. FDA is one of the widely used discriminant algorithms that seeks to separate different classes as much as possible for discriminant analysis or dimension reduction. It has been applied to face recognition, speech recognition, and handwriting recognition. The secure solutions are designed based on two basic secure building blocks that we have proposed—the Secure Matrix Multiplication protocol and the Secure Inverse of Matrix Sum protocol—which are in turn based on cryptographic techniques. We conducted experiments to evaluate the scalability of the proposed secure building blocks and overheads to achieve privacy when performing FDA.

1 Introduction

Data mining is a powerful tool to discover interesting, useful, and even hidden patterns that has been applied to various domain such as business intelligence, bioinformatics, and homeland security. While conventional data mining assumes that the data miner has full access rights to data that are collected from different sources or that are distributed among multiple parties, privacy or security issues render this assumption infeasible when the parties cannot be fully trusted, as some parties may have malicious intent. How to collaboratively perform data mining without compromising the data privacy of the participating parties has become an interesting topic of research in the data mining community.

Privacy-preserving data mining (PPDM) is a response from the data mining community to address data privacy issues. Approaches in PPDM are generally based on Secure Multi-party Computations (SMC) [12] and/or randomization techniques [1]. The former uses specialized, proven protocols to achieve various types of computation without losing data privacy. The latter introduces noise to the original private data to achieve security but lose accuracy. As the former approach achieves a higher degree of accuracy, we focus on SMC in this paper. To

T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 136–147, 2008.

date, various data mining algorithms have been enhanced to incorporate privacy preservation based on SMC techniques.

In machine learning and data mining, Fisher Discriminant Analysis (FDA) is one of the widely used discriminant algorithms that seeks to find directions so that data in the same classes are projected near to each other while ones in different classes are projected as far as possible for classification or dimension reduction. It has wide applications in face recognition [13], speech recognition [11], and digit recognition [2]. In this paper, we enhance Fisher Discriminant Analysis to incorporate the privacy-preserving feature. To the best of our knowledge, there has not been any work that extends privacy preservation to FDA.

Our contributions in this paper are summarized as follows:

1. We propose two protocols—the Secure Matrix Multiplication protocol and the Secure Inverse of Matrix Sum protocol—as secure basic building blocks for privacy-preserving FDA. The underlying algorithms of these protocols are novel and more secure than those by Du *et al.* [3].
2. Based on the two secure building blocks, we propose protocol for privacy-preserving FDA over horizontally and vertically partitioned data.

We have evaluated the computational complexity and scalability of the proposed protocols both analytically and empirically and show that the protocols are efficient and scalable for small to medium size data. We also addressed some specific implementation issues such as methods to handle real numbers and negative numbers in cryptography. We believe this work is significant as it serves as a guide to the investigation of extending data privacy preservation to related methods such as Principal Component Analysis, Independent Component Analysis, and so on.

The organization of this paper is as follows: In Section 2, we present an overview of background knowledge about linear FDA and related work. Section 3 proposes two secure building blocks of matrix computation. We also present protocols for Privacy-Preserving FDA (PPFDA) over horizontally partitioned data and vertically partitioned data in Section 4. In Section 5, we perform experiments to evaluate the proposed secure building blocks and protocols. Finally, Section 6 concludes the paper.

2 Background and Related Work

2.1 Linear Fisher Discriminant Analysis

Fisher Discriminant Analysis (FDA) as introduced by Fisher [5] seeks to separate different classes as much as possible using some criterion function (Eq. 1). As the technique of applying FDA on a two-class dataset is used repeatedly for the analysis of any pairs of data in a multi-class dataset, we focus on the two-class problem using FDA in this paper. It is non-trivial to extend the two-class problem approach to multi-class problems. This will be part of our future work. This section provides an overview of background knowledge about linear FDA.

We present the conventional mathematical model of linear FDA for two-class data [4]. Suppose we have a set of two-class n data samples of d dimensions: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where $\mathbf{x}_i = [x_{1,i}, x_{1,i}, \ldots, x_{d,i}]^T$ such that n_1 samples are in the subset $\varphi_1 = \{\mathbf{x}_1^1, \mathbf{x}_2^1 \ldots, \mathbf{x}_{n_1}^1\}$ and n_2 samples are in the subset $\varphi_2 = \{\mathbf{x}_1^2, \mathbf{x}_2^2 \ldots, \mathbf{x}_{n_2}^2\}$, $n_1 + n_2 = n$. Assuming that column vector $\mathbf{w}$ is the direction of the projection from $\mathbf{X}$ to $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$, we have $\mathbf{y} = \mathbf{w}^T\mathbf{X}$. The d-dimensional sample mean $\mathbf{m}_i$ for class i is $\mathbf{m}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}\mathbf{x}_j^i$.

Fisher Discriminant Analysis aims to maximize between-class separability and minimize within-class variability. Formally, the criterion function in Eq. 1 is to be maximized for the function $\mathbf{w}^T\mathbf{X}$:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} \tag{1}$$

where

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$$

$$\mathbf{S}_W = \sum_{i=1,2}\sum_{j=1}^{n_i}(\mathbf{x}_j^i - \mathbf{m}_i)(\mathbf{x}_j^i - \mathbf{m}_i)^T \tag{2}$$

is the *within-class scatter matrix*.

The objective of FDA is to find a projection vector $\mathbf{w}$ such that $J(\mathbf{w})$ in Eq. 1 is a maximum. The solution for such $\mathbf{w}$ can be obtained by differentiating $J(\mathbf{w})$ with respect to $\mathbf{w}$ yielding

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \tag{3}$$

We note that only the direction, not the length of $\mathbf{w}$, is important.

To incorporate the privacy-preserving feature to linear FDA, the challenge is to securely compute $\mathbf{S}_W^{-1}$ and $\mathbf{m}_1 - \mathbf{m}_2$ so that $\mathbf{w}$ can be securely computed. Clearly, what we need is a method to perform matrix multiplication and matrix inverse securely. In Section 4, we propose a secure approach to address the problem.

2.2 Secure Building Blocks

Various data mining algorithms have been enhanced to incorporate privacy preservation, including classification using decision tree [12], association rule mining [16], clustering using k-means [10], and so on. Recently, the approach has been extended to several machine learning algorithms such as linear regression [3], gradient descent methods [17], self-organizing maps [8], and genetic algorithms [7]. Many of these privacy-enabled algorithms rely on secure building blocks to enforce privacy. Secure building blocks are basic common operations that underly many algorithms. Examples include secure sum, secure comparison, secure scalar product, secure matrix multiplication, and so on.

Fisher Discriminant Analysis—the focus of this paper—requires two secure building blocks: Secure matrix multiplication and secure inverse of matrix sum.

Du *et al.* [3] has proposed a secure protocol for secure matrix multiplication using linear algebraic methods. It uses a random and invertible matrix to disguise the original matrices to achieve privacy. For security, a concept called "k-secure" was introduced to generate the random matrix. Assuming that Party B wants to attack private matrix $\mathbf{A}$ of Party A, a k-secure matrix $\mathbf{M}$ (jointly generated by both parties) means that (1) any equation from $\mathbf{MA}$ includes at least $k+1$ unknown elements of $\mathbf{A}$, and (2) any k combined equations include at least $2k$ unknown elements of $\mathbf{A}$. Therefore, it is impossible to know any elements of matrix $\mathbf{A}$ as there are infinite possible solutions due to insufficient equations.

An issue with Du's approach is that constructing such a matrix is a complex process [3]. More importantly, Du's approach may have a security problem. If Party A and the same Party B or different Party Bs (a group of colluding parties) perform secure matrix multiplication more than once, more $\mathbf{M}$s (more equations) are available for attacking the fixed unknown elements matrix $\mathbf{A}$. In response to this problem, we propose another more secure and efficient protocol for matrix multiplication in this paper.

3 Secure Building Blocks

In this section, we propose the Secure Matrix Multiplication protocol and Secure Inverse of Matrix Sum protocol to support the secure computation of Eq. 3, which we have identified to be the key to incorporating privacy preservation in FDA. Our proposed protocols are based on cryptographic techniques and are improvements over existing protocols [3] for secure matrix multiplication and inverse of matrix sum.

3.1 Secure Matrix Multiplication

Parties A and B each hold private $d \times N$ matrix $\mathbf{A}$ and private $N \times n$ matrix $\mathbf{B}$ respectively. They want to securely compute matrix multiplication so that at the end of the computation, party A and B each only holds a portion of the product matrix $\mathbf{M}^a$ and $\mathbf{M}^b$ respectively such that their matrix sum $\mathbf{M}^a + \mathbf{M}^b = \mathbf{AB}$ is the desired product matrix, which is unknown to both parties.

Given any $m \times n$ matrix $\mathbf{H}$, its ith row vector $\mathbf{h}(i,:) = (h_{i,1}, h_{i,2}, \ldots, h_{i,n})$ and jth column vector $\mathbf{h}(:,j) = (h_{1,j}, h_{2,i}, \ldots, h_{m,j})$. By definition of matrix multiplication $\mathbf{M}$=$\mathbf{AB}$, we have

$$\mathbf{M} = \mathbf{AB} = \begin{bmatrix} \mathbf{a}(1,:)\cdot\mathbf{b}(:,1) & \mathbf{a}(1,:)\cdot\mathbf{b}(:,2) & \cdots & \mathbf{a}(1,:)\cdot\mathbf{b}(:,n) \\ \mathbf{a}(2,:)\cdot\mathbf{b}(:,1) & \mathbf{a}(2,:)\cdot\mathbf{b}(:,2) & \cdots & \mathbf{a}(2,:)\cdot\mathbf{b}(:,n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}(d,:)\cdot\mathbf{b}(:,1) & \mathbf{a}(d,:)\cdot\mathbf{b}(:,2) & \cdots & \mathbf{a}(d,:)\cdot\mathbf{b}(:,n) \end{bmatrix}$$

Clearly, each element of $\mathbf{M}$ above is a scalar product of two vectors. To securely perform the matrix multiplication $\mathbf{AB}$, we may apply the Secure Scalar Product

Protocol 1. Secure Matrix Multiplication Protocol

Input: Party A has private $d \times N$ matrix $\mathbf{A}$ and Party B has private $N \times n$ matrix $\mathbf{B}$.
Output: Party A obtains private matrix $\mathbf{M}^a$ and Party B obtains private matrix $\mathbf{M}^b$ such that their sum $\mathbf{M}^a + \mathbf{M}^b = \mathbf{AB}$ yields the product matrix.

1: **for** $i = 1$ to d **do**
2: **for** $j = 1$ to n **do**
3: Party A and Party B securely compute the scalar product of vector $\mathbf{a}(i,:)$ and vector $\mathbf{b}(:,j)$. At the end, Party A and Party B each hold a private value $m^a_{i,j}$ and $m^b_{i,j}$ respectively. Part A designates $\mathbf{M}^a_{i,j} = m^a_{i,j}$ and Party B designates $\mathbf{M}^b_{i,j} = m^b_{i,j}$.
4: **end for**
5: **end for**

protocol [6] so that each scalar product is the sum of two portions as follows:

$$\mathbf{AB} = \begin{bmatrix} m^a_{1,1} + m^b_{1,1} & m^a_{1,2} + m^b_{1,2} & \cdots & m^a_{1,n} + m^b_{1,n} \\ m^a_{2,1} + m^b_{2,1} & m^a_{2,2} + m^b_{2,2} & \cdots & m^a_{2,n} + m^b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m^a_{d,1} + m^b_{d,1} & m^a_{d,2} + m^b_{d,2} & \cdots & m^a_{d,n} + m^b_{d,n} \end{bmatrix} = \mathbf{M}^a + \mathbf{M}^b$$

In this way, we securely obtain the matrix multiplication (which is unknown to both parties) as the sum of two private portions $\mathbf{M}^a$ and $\mathbf{M}^b$ held by Party A and B respectively. The details are shown in Protocol 1.

This method is more straightforward and less complex than the secure matrix multiplication protocol by Du *et al.* [3]. Moreover, the execution of secure scalar product of each matrix element can be performed concurrently to increase efficiency. In Section 5, we show that the approach is efficient for computing the product of two small and medium size matrices.

3.2 Secure Inverse of Matrix Sum

Party A and B each hold a private $d \times d$ matrix $\mathbf{A}$ and $\mathbf{B}$ respectively. They want to securely compute the inverse of $\mathbf{A} + \mathbf{B}$. At the end of the secure computation, Party A and B each only holds a portion of the inverse matrix $\mathbf{M}^a$ and $\mathbf{M}^b$ respectively such that their sum $\mathbf{M}^a + \mathbf{M}^b = (\mathbf{A} + \mathbf{B})^{-1}$; the inverse matrix is not known to both parties.

The steps to securely perform the inverse of matrix sum by two parties are shown in Protocol 2. In Steps 1 to 3, Party B uses a random, non-singular matrix $\mathbf{P}$ to hide its private matrix $\mathbf{B}$ before sending it to Party A. In Steps 4 and 5, both both parties securely compute the inverse of $(\mathbf{A}+\mathbf{B})\mathbf{P}$ and then the product $\mathbf{P}(\mathbf{P}^{-1}(\mathbf{A} + \mathbf{B})^{-1})$, essentially eliminating the random matrix $\mathbf{P}$ in the process. This yields the desired result $(\mathbf{A} + \mathbf{B})^{-1}$ in the form of two private portions $\mathbf{M}^a$ and $\mathbf{M}^b$ held by each party respectively.

In the case when the sum matrix $\mathbf{A}+\mathbf{B}$ is singular, a simple perturbation can be introduced to the sum matrix to make it non-singular. For instance, the

Protocol 2. Secure Inverse of Matrix Sum Protocol

Input: Party A has private $d \times d$ matrix $\mathbf{A}$ and Party B has private $d \times d$ matrix $\mathbf{B}$.

Output: Party A obtains private matrix $\mathbf{M}^a$ and Party B obtains private matrix $\mathbf{M}^b$ such that their sum $\mathbf{M}^a + \mathbf{M}^b = (\mathbf{A} + \mathbf{B})^{-1}$ yields the inverse of the sum of their private matrices.

1: Party B randomly generates a non-singular $d \times d$ matrix $\mathbf{P}$.
2: Party A and Party B jointly perform secure matrix multiplication (using Protocol 1) to compute $\mathbf{AP}$, at the end of which, Party A and Party B each obtain $\mathbf{S}^a$ and $\mathbf{S}^b$ respectively such that $\mathbf{S}^a + \mathbf{S}^b = \mathbf{AP}$.
3: Party B computes $\mathbf{S}^b + \mathbf{BP}$ and sends it to Party A.
4: Party A computes $\mathbf{S}^a + \mathbf{S}^b + \mathbf{BP}$; i.e., $(\mathbf{A}+\mathbf{B})\mathbf{P}$, and then its inverse $\mathbf{P}^{-1}(\mathbf{A}+\mathbf{B})^{-1}$.
5: Party B and Party A jointly perform secure matrix multiplication (using Protocol 1) on $\mathbf{P}$ and $\mathbf{P}^{-1}(\mathbf{A}+\mathbf{B})^{-1}$, at the end of which, Party A and Party B each hold private portions $\mathbf{M}^b$ and $\mathbf{M}^a$ respectively such that $\mathbf{M}^a + \mathbf{M}^b = \mathbf{P}(\mathbf{P}^{-1}(\mathbf{A} + \mathbf{B})^{-1}) = (\mathbf{A} + \mathbf{B})^{-1}$.

perturbation method proposed by Hong and Yang [9] can be used to stabilize $\mathbf{A} + \mathbf{B}$ by adding a small perturbation matrix to $\mathbf{A}$ or $\mathbf{B}$.

In contrast to the secure inverse of matrix sum protocol by Du *et al.* [3], Protocol 2 is more efficient and accurate as it uses only one random matrix $\mathbf{P}$ instead of two matrices in Du's protocol. Clearly, less (one random matrix less) algebraic operations yields more accurate computations results as less errors are introduced due to roundoff errors.

4 Privacy-Preserving FDA

4.1 PPFDA over Horizontally Partitioned Data

In this scenario, we have n data samples of d dimensions held by two parties. Let Party A hold the first n_1 data samples and Party B hold the remaining n_2 data samples; $n = n_1 + n_2$.

In Protocol 3, we show how $\mathbf{m}_1 - \mathbf{m}_2$ and $\mathbf{S}_W^{-1}$ can be securely computed so as to yield $\mathbf{w}$ in a secure manner. In addition to using Protocols 1 and 2, we also make use of the *random shares* technique by Jagannathan and Wright [10]. In this technique, all *numerical* intermediate results are splitted into two random portions where each party holds one portion so that neither party is able to speculate anything about the intermediate results using only its private portion.

In Step 1, we show how $\mathbf{m}_1 - \mathbf{m}_2$ can be splitted into two random portions. As Party A holds n^a data samples (with n_i^a data samples of class i) and Party B holds n^b data samples (with n_i^b data samples of class i), $n_i = n_i^a + n_i^b$, the mean vector of class i as computed by Party A using only its private data samples is $\mathbf{m}_i^a$. Likewise, the mean vector of class ii computed by Party B using its private data samples is $\mathbf{m}_i^b$. Hence, we have

Protocol 3. PPFDA over Horizontally Partitioned Data

Input: Party A has n^a private d-dimensional data samples. Party B has n^b private d-dimensional data samples.

Output: Party A and Party B securely compute a projection vector $\mathbf{w}$ for the data samples held by them.

1: Party A computes $\mathbf{t}^a = (n_1^a/n_1)\mathbf{m}_1^a - (n_2^a/n_2)\mathbf{m}_2^a$. Party B computes $\mathbf{t}^b = (n_1^b/n_1)\mathbf{m}_1^b - (n_2^b/n_2)\mathbf{m}_2^b$.
2: Party A sets $\mathbf{S}_W^a = \mathbf{0}$ and Party B sets $\mathbf{S}_W^b = \mathbf{0}$.
3: **for** $i = 1$ to 2 **do**
4: **for** $j = 1$ to n_i **do**
5: **if** ($\mathbf{x}_j^i$ is held by Party A) **then**
6: $\mathbf{u}^a = \mathbf{x}_j^i - (n_i^a/n_i)\mathbf{m}_i^a$ and $\mathbf{u}^b = -(n_i^b/n_i)\mathbf{m}_i^b$
7: **else**
8: $\mathbf{u}^a = -(n_i^a/n_i)\mathbf{m}_i^a$ and $\mathbf{u}^b = \mathbf{x}_j^i - (n_i^b/n_i)\mathbf{m}_i^b$
9: **end if**
10: Using Eq. 4, $\mathbf{M}^a + \mathbf{M}^b = (\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T$
11: Update $\mathbf{S}_W^a = \mathbf{S}_W^a + \mathbf{M}^a$ and $\mathbf{S}_W^b = \mathbf{S}_W^b + \mathbf{M}^b$
12: **end for**
13: **end for**
14: Both parties jointly perform secure inverse of matrix sum (Protocol 2) to obtain $\mathbf{S}^a + \mathbf{S}^b = (\mathbf{S}_W^a + \mathbf{S}_W^b)^{-1}$.
15: Both parties jointly perform secure matrix multiplication (Protocol 1) to obtain $\mathbf{S}^a\mathbf{t}^b$ and $\mathbf{S}^b\mathbf{t}^a$; projection vector $\mathbf{w} = \mathbf{S}^a\mathbf{t}^a + \mathbf{S}^a\mathbf{t}^b + \mathbf{S}^b\mathbf{t}^a + \mathbf{S}^b\mathbf{t}^b$ (Eq. 6) may now be computed.

Notations: n_1^a and n_2^a refer to the number of data samples of classes 1 and 2 respectively held by Party A; n_1^b and n_2^b refer to the number of data samples of classes 1 and 2 respectively held by Party B.

$$\begin{aligned}\mathbf{m}_1 - \mathbf{m}_2 &= \frac{n_1^a\mathbf{m}_1^a + n_1^b\mathbf{m}_1^b}{n_1} - \frac{n_2^a\mathbf{m}_2^a + n_2^b\mathbf{m}_2^b}{n_2} \\ &= \left(\frac{n_1^a}{n_1}\mathbf{m}_1^a - \frac{n_2^a}{n_2}\mathbf{m}_2^a\right) + \left(\frac{n_1^b}{n_1}\mathbf{m}_1^b - \frac{n_2^b}{n_2}\mathbf{m}_2^b\right) \\ &= \mathbf{t}^a + \mathbf{t}^b\end{aligned}$$

Next, Steps 2 to 13 securely compute $\mathbf{S}_W = \sum_{i=1,2}\sum_{j=1}^{n_i}(\mathbf{x}_j^i - \mathbf{m}_i)(\mathbf{x}_j^i - \mathbf{m}_i)^T$ (Eq. 2). The secure manner to compute $\mathbf{S}_W$ is to obtain two portion matrices $\mathbf{S}_W^a$ and $\mathbf{S}_W^b$ each held by Party A and Party B respectively. This is performed using the two **for** loops as shown in the protocol.

In Step 5, if $\mathbf{x}_j^i$ belongs to Party A, then we have

$$\begin{aligned}\mathbf{x}_j^i - \mathbf{m}_i &= \left(\mathbf{x}_j^i - \frac{n_i^a}{n_i}\mathbf{m}_i^a\right) + \left(-\frac{n_i^a}{n_i}\mathbf{m}_i^b\right) \\ &= \mathbf{u}^a + \mathbf{u}^b\end{aligned}$$

which yields $(\mathbf{x}_j^i - \mathbf{m}_i)(\mathbf{x}_j^i - \mathbf{m}_i)^T = (\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T$. The same process is performed if $\mathbf{x}_j^i$ belongs to Party B.

Step 10 shows the secure manner to split the resultant product matrix of two vectors $(\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T$ into two portions such that

$$\mathbf{M} = (\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T = \mathbf{M}^a + \mathbf{M}^b \tag{4}$$

The element $m_{i,j}$ of matrix $\mathbf{M}$ is computed as follows:

$$\begin{aligned} m_{i,j} &= (u_i^a + u_i^b)(u_j^a + u_j^b) \\ &= u_i^a \times u_j^a + \begin{bmatrix} u_i^a \\ u_j^a \end{bmatrix} \cdot \begin{bmatrix} u_j^b \\ u_i^b \end{bmatrix} + u_i^b \times u_j^b \\ &= m_{i,j}^a + m_{i,j}^b \end{aligned} \tag{5}$$

After securely computing the scalar product of vectors in Eq. 5, each element of matrix $\mathbf{M}$ is splitted into two portions. Hence, the matrix $\mathbf{M}$ is splitted into two private matrices. Overall, $\mathbf{S}_W$ is securely splitted into two private portions $\mathbf{S}_W^a$ and $\mathbf{S}_W^b$.

Using Protocol 2, $(\mathbf{S}_W)^{-1} = \left(\mathbf{S}_W^a + \mathbf{S}_W^b\right)^{-1}$ can be securely splitted into $\mathbf{S}^a$ and $\mathbf{S}^b$ such that $(\mathbf{S}_W)^{-1} = \mathbf{S}^a + \mathbf{S}^b$. Therefore

$$\begin{aligned} \mathbf{w} &= (\mathbf{S}_W)^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \\ &= (\mathbf{S}^a + \mathbf{S}^b)(\mathbf{t}^a + \mathbf{t}^b) \\ &= \mathbf{S}^a\mathbf{t}^a + \mathbf{S}^a\mathbf{t}^b + \mathbf{S}^b\mathbf{t}^a + \mathbf{S}^b\mathbf{t}^b \end{aligned} \tag{6}$$

Using Protocol 1, we securely compute $\mathbf{S}^a\mathbf{t}^b$ and $\mathbf{S}^b\mathbf{t}^a$. Thus, we are able to securely compute $\mathbf{w}$.

Analysis: Two parities are assumed to be semi-honest who strictly follow the protocol but collect all intermediate results during the execution of protocols to attack the private data of honest parties. As we observe, Protocol 3 applies two main secure building blocks: Secure Matrix Multiplication protocol and Secure Inverse of Matrix Sum protocol. Both protocols depend on the Secure Scalar Product protocol that is provably secure [6]. Based on random share technique, we actually split all the intermediate results into two random shares (portions) except the final $\mathbf{w}$ in Protocol 3. The private variables of one party are protected by the equivalent numbers of random portions known by itself only. Therefore we claim data privacy of honest parties are preserved.

We derive computational complexity of Protocol 3 here. As in Steps 3 to 13, the Secure Scalar Product protocol is invoked once to compute the scalar product of two vectors (2×1) (in Eq. 5), then one element of matrix $\mathbf{M}$ $(d \times d)$ is securely split. As we know, there are $n_1 + n_2 = n$ data. Therefore, the Secure Scalar Product protocol is invoked $n \times d^2$ times in Steps 3 to 13 for computing the scalar product of two vectors (2×1).

In Step 14, the Secure Inverse of Matrix Sum protocol is invoked once for splitting $(\mathbf{S}_W^a + \mathbf{S}_W^b)^{-1}$ $(d \times d)$. It requires to run the Secure Matrix Multiplication protocol twice (Step 2 and 5 in Protocol 2). In Step 14, the Secure Matrix

Multiplication protocol is invoked twice for splitting items $\mathbf{S}^a\mathbf{t}^b$ $(d \times 1)$ and $\mathbf{S}^b\mathbf{t}^a$ $(d \times 1)$ securely. In the Secure Matrix Multiplication protocol, it requires to perform the Secure Scalar Product protocol once to split one element of the desired matrix. The overall number of invoking Secure Scalar Product protocol in Step 14 and 15 is $(2d^2 + 2d)$ for computing the scalar product of two vectors $(d \times 1)$.

Therefore, the overall computational complexity is $O(nd^2 + d^3)$ as the computational complexity of the Secure Scalar Product protocol is $O(\tau)$ for two vectors of length τ [6].

The communication of Protocol 3 between two parties mainly comes from depends on Secure Scalar Product protocol invoked in the protocol. Based on the analysis above, the the communication complexity of Protocol 3 depends on the overall number of the Secure Scalar Product protocol invoked, which is $O(nd^2 + d^3)$ as the communication complexity of the Secure Scalar Product protocol is $O(\tau)$ for two vectors of length τ [6].

In Section 5, we experimentally evaluate the efficiency and scalability of the secure building blocks.

4.2 PPFDA over Vertically Partitioned Data

In this scenario, d dimensions of data are distributed between two parties. Party A holds d_1 dimensions and Party B holds d_2 dimensions; $d = d_1 + d_2$. We show how $\mathbf{w}$ can be securely computed in such a scenario.

In vertically partitioned data, we assume the first d_1 dimensions of data sample $\mathbf{x} = [x_1, x_2, \ldots, x_d]$ are held by Party A: $\mathbf{x}^a = [x_1, x_2, \ldots, x_{d_1}]^T$ and the remaining d_2 dimensions of $\mathbf{x}$ are held by Party B: $\mathbf{x}^b = [x_{d_1+1}, x_{d_1+2}, \ldots, x_{d_1+d_2}]^T$. We show that Party A and Party B may *extend* their vertical data partitions with empty dimensions so that both parties have d dimensional partitions. In this way, the problem of vertically partitioned data is transformed to a horizontally partitioned problem so that the method in Section 4.1 can be applied to securely compute $\mathbf{w}$.

The transformation is as follows: For each d_1 dimension data sample $\mathbf{x}^a$ of Party A, additional d_2 zeroes can be appended so that the data sample has d dimension:

$$(\mathbf{x}^a)' = [x_1, x_2, \ldots, x_{d_1}, \overbrace{0, 0, \ldots, 0}^{d_2}]^T$$

Likewise, data samples of Party B can be prepended with d_1 zeroes to become d dimensional:

$$(\mathbf{x}^b)' = [\overbrace{0, 0, \ldots, 0}^{d_1}, x_{d_1+1}, x_{d_1+2}, \ldots, x_{d_1+d_2}]^T$$

After the transformation, we have a total of $2n$ data samples of d dimensions rather than n data samples of d_1 held by Party A and n data samples of d_2 held by Party B.

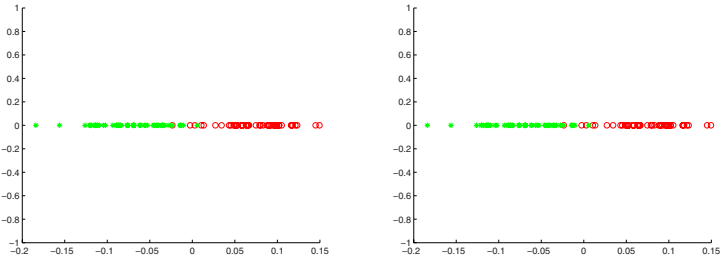

Fig. 1. Accuracy comparison of FDA without and with privacy

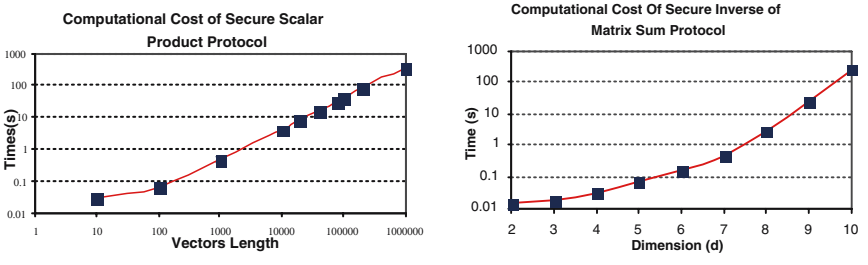

Fig. 2. Scalability of the Secure Scalar Product Protocol and Secure Inverse of Matrix Sum

5 Experiments

In this section, we discuss the implementation issues and evaluate the performance of the proposed protocols. All protocols were implemented in the C# language running under Microsoft Visual Studio 2005 environment. All experiments are performed on the Window XP operating system with 3.40GHz CPU and 1GB memory. As network performance mainly depends on the network speed and physical distance of two parties, we simply implemented parties as threads that exchange data directly by shared memory.

The dataset used is the Iris Plants Database from the UCI Machine Learning Depository. There are 150 data samples in three classes: "Iris Setosa", "Iris Versicolour", and "Iris Virginica" . As the latter two classes are not linearly separable, we select them as our analysis data. There are 4 numeric predictive attributes: "sepal length", "sepal width", "petal length", and "petal width".

The Paillier cryptosystem [14] was selected as our choice in the implementation. As the Paillier cryptosystem only encrypts non-negative integers, we have to deal with issues when real numbers and negative numbers occur. For real numbers, two parties multiply some large constants (e.g., 1000) to transform the real numbers to integers. We remove the effects of the constants by dividing the (intermediate) results by the constants. For negative numbers, the basic property of congruence $a+kn = a \bmod n$ is applied to transform negative integer a to positive integers by adding multiples of n.

Table 1. Efficiency analysis of Secure Inverse of Matrix Sum protocol

Dimension ($d \times d$)	Secure Inverse of Matrix Sum	Matrix Inverse	Overhead from Secure Matrix Multiplication
5	0.069516s	0.015586s	0.05393s
6	0.15586s	0.046758s	0.109102s
7	0.436408s	0.249376s	0.187032s
8	2.72755s	2.228798s	0.498752s
9	24.095956s	21.867158	2.228798s
10	244.8105s	240.055572s	4.754928s

Accuracy: To show the accuracy of performing FDA with privacy preservation using Protocol 3, we evaluated horizontally partitioned data where data instances of data set are uniformly distributed between two parties. The first figure in Fig. 1 was obtained by performing FDA using MATLAB. The second figure was obtained by Protocol 3. We clearly observe that accuracy is not reduced when we preserve the data privacy of the participant parties.

Scalability: We investigate the scalability of the two protocols proposed in this paper. For the Secure Matrix Multiplication protocol, we observe that the bulk of its operations are secure scalar products. Hence, we evaluated the scalability of the Secure Scalar Product protocol as shown in the first figure in Fig. 2. The running time is linear to the length of vectors as expected. Some random numbers in our implementation were generated offline. The time for two vectors of length 100,000 was estimated at 41 seconds, which is sufficiently low for small and medium data sets. The second figure in Fig. 2 shows the efficiency of the Secure Inverse of Matrix Sum protocol. We observe that the time to execute the protocol for more than 10×10 dimensions matrices becomes impractical. From Table 1, it is shown that the matrix inverse algorithm we used is time consuming due to the computation of matrix inverse and not due to overhead of the Secure Matrix Multiplication protocol. In our experiment, we use adjoint method [15] to perform matrix inverse as follows: $\mathbf{A}^{-1} = (1/\det)\mathbf{A}(\text{adjoint of } \mathbf{A})$ which is very computationally slow, comparing with other methods, such as Gauss-Jordan elimination and LU decomposition.

In these experiments, we only evaluated privacy-preserving FDA over horizontally partitioned data for low dimension (4×4). To apply the proposed protocol to higher dimension data would be part of our future work.

6 Conclusions

In this paper, we have proposed the privacy-preserving version of Fisher Discriminant Analysis over horizontally and vertically partitioned data. We have also proposed two basic secure building blocks for matrix computation: the Secure Matrix Multiplication protocol and Secure Inverse of Matrix Sum protocol. Finally, we have conducted experiments to demonstrate the scalability of the proposed secure building blocks and overheads to achieve the privacy when

performing FDA. Our future work includes applying the proposed protocol to high-dimensional data and extending the proposed protocols to multiple parties.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the ACM SIGMOD, Dallas, Texas, United States, pp. 439–450 (2000)
2. Berkes, P.: Handwritten digit recognition with nonlinear fisher discriminant analysis. In: Proceedings of the International Conference on Artificial Neural Networks, pp. 285–287 (2005)
3. Du, W., Han, Y., Chen, S.: Privacy-preserving multivariate statistical analysis: Linear regression and classification. In: Proceedings of the 4th SIAM International Conference on Data Mining, Florida, April 22–24, 2004, pp. 222–233 (2004)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. Wiley-Interscience, Chichester (2000)
5. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
6. Goethals, B., Laur, S., Lipmaa, H., Mielikainen, T.: On private scalar product computation for privacy-preserving data mining. In: Proceedings of the 7th Annual International Conference in Information Security and Cryptology, pp. 104–120
7. Han, S., Ng, W.K.: Privacy-Preserving Genetic Algorithms for Rule Discovery. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 407–417. Springer, Heidelberg (2007)
8. Han, S., Ng, W.K.: Privacy-preserving self-organizing map. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 428–437. Springer, Heidelberg (2007)
9. Hong, Z.-Q., Yang, J.-Y.: Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recognition 24(4), 317–324 (1991)
10. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: Proceedings of the 8th ACM SIGKDD, Chicago, Illinois, USA, pp. 593–599 (2005)
11. Katz, M., Meier, H.G., Dolfing, H., Klakow, D.: Robustness of linear discriminant analysis in automatic speech recognition. In: Proceedings of the 16th International Conference on Pattern Recognition, pp. 371–374 (2002)
12. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 36–53. Springer, Heidelberg (2000)
13. Lu, J., Plataniotis, K.N., Venetsanopoulos, A.N.: Face recognition using kernel direct discriminant analysis algorithms. IEEE Transactions on Neural Networks 14(1), 117–126 (2003)
14. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
15. Strang, G.: Linear algebra and its applications. Thomson, Brooks/Cole (2006)
16. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: Proceedings of the 8th ACM SIGKDD, Edmonton, Alberta, Canada, July 23-26, 2002, pp. 639–644 (2002)
17. Wan, L., Ng, W.K., Han, S., Lee, V.C.S.: Privacy-preservation for gradient descent methods. In: Proceedings of the 13th ACM SIGKDD, San Jose, California, USA, August 2007, pp. 775–783 (2007)