FRiPAL: Face Recognition in Privacy Abstraction Layer

Di Zhuang*, Sen Wang†, and J. Morris Chang‡
Department of Electrical Engineering, University of South Florida, Tampa, Florida 33620
Email: *dizhuang@mail.usf.edu, †senwang@mail.usf.edu, ‡chang5@usf.edu

***Abstract*—Data-driven mobile applications are becoming increasingly popular in civilian and law enforcement. RapidGather, for instance, is an smartphone application that collects data from individual, and spreads rapid emergency responses. Image data is widely used in such applications, and machine learning methods could be utilized to analyze the image data. However, people would hesitate to share the data without protecting their privacy. In this paper, we propose to utilize dimensionality reduction techniques for privacy-preserving machine learning in face recognition for the image data. To demonstrate the proposed approach, we implement a client server system, FRiPAL. With extensive experiments, we show that FRiPAL is efficient, and could preserve the privacy of data owners while maintaining the utility for data users.**

I. Introduction

Modern data-driven applications are becoming increasingly popular in civilian and law enforcement. Such applications collect data from the smartphones, analyze the data at backend systems, and help people to make decisions. RapidGather [1], for instance, is a data-driven emergency response application that collects different types of data from smartphones, and spreads rapid emergency response to citizens and authorities. However, smartphone users might hesitate to share their data, if RapidGather could not protect the data privacy properly.

Gathering and analyzing photos rapidly is of great importance in emergency events. For instance, in the Boston Marathon bombing scenario (a potential RapidGather use case), even if information transmission immediately through Internet, social media and news report, it still took several days for authorities to gather photos from smartphone users who were in that area, and pore through thousands of photos to identify the suspects. We come up with a privacy-preserving mechanism that could motivate the data owners to share their photos with the authorities, and the authorities could query photos from the crowd around the scene rapidly, and recognize the wanted suspects effectively.

In this paper, we propose a privacy-preserving machine learning framework for face recognition for the image data in the RapidGather application. Machine learning is an important tool to model the appearance of faces and to classify them. Eigenface [2] and Fisherface [3] have been utilized for a long time for face recognition. However, those traditional methods do not consider privacy issues. As the demand for privacy increasing, privacy-preserving machine learning becomes an emerging area. To date, a few approaches rely on cryptographic protocols (e.g. homomorphic [4] or commutative encryption [5]) or data perturbation (e.g. random projection [6]) techniques have been proposed. However, the cryptography-based approaches suffer from low efficiency, due to high computation and communication cost. The data perturbation based approaches suffer from low accuracy, due to the loss of useful information. Therefore, instead of utilizing cryptography or data perturbation techniques, we propose to utilize dimensionality reduction techniques for privacy-preserving machine learning. These techniques can efficiently transform the raw data from the data owner to a new set of data before they are given to the data users. Without revealing the raw data, the transformation is irreversible.

We have implemented our methods in a system called FRiPAL, Face Recognition in Privacy Abstraction Layer, which is a privacy-preserving face recognition service design. RapidGather proposed [1] an architecture of Privacy-Enhanced Android (PE-Android) which is an extension of the current Android OS with new privacy features. One of the most important components in PE-Android is the Privacy Abstraction Layer (PAL), which is defined as a wrapper of the low level PE-Android services that allows the developers to develop privacy preserving applications in their traditional way. FRiPAL has been integrated into RapidGather as a privacy-preserving face recognition service for image data.

The contributions of this paper are as follows:

First. We propose to utilize dimensionality reduction techniques for privacy-preserving machine learning in face recognition. We demonstrate the proposed approach with three dimensionality reduction methods, including Principal Component Analysis (PCA) [7], Linear Discriminant Analysis (LDA) [3] and Discriminant Component Analysis (DCA) [8].

Second. We design and implement a privacy-preserving face recognition client server system, FRiPAL, which could preserve the privacy of data owners while maintaining the utility for data users.

Third. Extensive experiments have been conducted on three different public datasets to evaluate FRiPAL in terms of accuracy, privacy and efficiency. The accuracy results show that our system maintains the utility for face recognition. The privacy results illustrate that our system protects the privacy which motivates the data owners to submit photos. The efficiency results demonstrate that our system is efficient for practical usage.

The rest of this paper is organized as follows. Section II

*† The first two authors contributed equally to this work.

978-1-5090-5569-2/17/$31.00 ©2017 IEEE

presents the preliminaries of dimensionality reduction methods. Section III describes the privacy-preserving face recognition problem and our proposed solution. Section IV describes the system design of FRiPAL. Section V presents the experimental evaluation. Section VI presents the related works. Section VII presents the conclusion and future work.

II. Preliminaries

A. Privacy-preserving by Dimensionality Reduction

In machine learning, dimensionality reduction is a tool to transform the feature vector from a high dimension space to a low dimension space. It has been used to deal with: (a) over-fitting problems when the number of features far exceed the number of training samples, (b) performance degradation due to suboptimal search, and (c) high computational cost and power consumption resulting from high dimensional feature space. However, in this paper, we investigate the privacy preserving usage of dimensionality reduction.

Dimensionality reduction is more resilient to reconstruction attacks [6]. Since the mapping from the original feature vectors to a low dimensional subspace is a many-to-one mapping, it is impossible to determine the privately held features from the reduced feature vectors without knowing any of the original feature vectors, as there are infinite possible feature vectors which could lead to identical reduced feature vector. Therefore, by utilizing dimensionality reduction, the data privacy is preserved since this transformation is irreversible. In this paper, we utilize three dimensionality reduction methods, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Discriminant Component Analysis (DCA).

B. Principal Component Analysis (PCA)

Consider a training data set consisting of n m-dimensional vectors: $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\}$, where $\boldsymbol{x_i} \in \mathbb{R}^m$. Below shows the general steps of PCA:

1) Compute the m-dimensional mean vector $\boldsymbol{\mu}$ of the whole data set:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i} \tag{1}$$

2) Compute the scatter matrix $\bar{\boldsymbol{S}}$ (alternatively, the covariance matrix) of the whole data set:

$$\bar{\boldsymbol{S}} = \sum_{i=1}^{n} (\boldsymbol{x_i} - \boldsymbol{\mu})(\boldsymbol{x_i} - \boldsymbol{\mu})^T \tag{2}$$

3) Compute the eigenvectors $\{\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots, \boldsymbol{e_m}\}$ and corresponding eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ of scatter matrix $\bar{\boldsymbol{S}}$ through spectral decomposition, e.g. eigen decomposition.
4) Sort the eigenvectors by non-increasing eigenvalues and choose d eigenvectors with the largest eigenvalues to form a projection matrix $\boldsymbol{W_{pca}} \in \mathbb{R}^{m \times d}$, where each column is an eigenvector.
5) Transform each sample onto the new subspace:

$$\boldsymbol{x_i'} = \boldsymbol{W_{pca}^T} \times \boldsymbol{x_i} \tag{3}$$

where $\boldsymbol{x_i} \in \mathbb{R}^m$, and $\boldsymbol{x_i'} \in \mathbb{R}^d$.

$\boldsymbol{W_{pca}}$ is the PCA projection matrix. The parameter d determines the dimension of the subspace of the transformed data, and the signal power retained after dimensionality reduction. For instance, suppose the original feature vectors have full signal power $\sum_{i=1}^{m} \lambda_i$, the transformed data has signal power $\sum_{i=1}^{d} \lambda_i$, and signal power $\sum_{i=d+1}^{m} \lambda_i$ has been irreversibly lost. We consider more privacy is preserved, as more signal power losing. Therefore, d could be utilized to control the level of privacy.

C. Linear Discriminant Analysis (LDA)

Consider a k-class training data set consisting of n m-dimensional vectors $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\}$, where $\boldsymbol{x_i} \in \mathbb{R}^m$. Each training sample $\boldsymbol{x_i}$ associates with a class label y_i indicating its belonging to one of the k classes $C_1, C_2, \ldots, C_k$. Each class C_j contains n_j training samples in this data set. Below shows the general steps of LDA:

1) Compute the total mean vector $\boldsymbol{\mu} \in \mathbb{R}^m$, and the class mean vector $\boldsymbol{\mu_j} \in \mathbb{R}^m$, $j = 1, 2, \ldots, k$:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i} \qquad \boldsymbol{\mu_j} = \frac{1}{n_j}\sum_{y_i \in C_j} \boldsymbol{x_i} \tag{4}$$

2) Compute the between-class scatter matrix $\boldsymbol{S_B}$ and the within-class scatter matrix $\boldsymbol{S_W}$:

$$\boldsymbol{S_B} = \sum_{j=1}^{k} n_j(\boldsymbol{\mu_j} - \boldsymbol{\mu})(\boldsymbol{\mu_j} - \boldsymbol{\mu})^T \tag{5}$$

$$\boldsymbol{S_W} = \sum_{j=1}^{k} \sum_{y_i \in C_j} (\boldsymbol{x_i} - \boldsymbol{\mu_j})(\boldsymbol{x_i} - \boldsymbol{\mu_j})^T \tag{6}$$

3) Compute the eigenvectors $\{\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots, \boldsymbol{e_m}\}$ and corresponding eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ of scatter matrix $\boldsymbol{S_W^{-1}}\boldsymbol{S_B}$ through spectral decomposition, e.g. eigen decomposition.
4) Sort the eigenvectors by non-increasing eigenvalues and choose d eigenvectors with the largest eigenvalues to form a projection matrix $\boldsymbol{W_{lda}} \in \mathbb{R}^{m \times d}$, where each column is an eigenvector.
5) Transform each sample onto the new subspace:

$$\boldsymbol{x_i'} = \boldsymbol{W_{lda}^T} \times \boldsymbol{x_i} \tag{7}$$

where $\boldsymbol{x_i} \in \mathbb{R}^m$, and $\boldsymbol{x_i'} \in \mathbb{R}^d$.

$\boldsymbol{W_{lda}}$ is the LDA projection matrix. Unlike PCA, LDA can reduce the m dimensionality at most to $k - 1$, because of the k-discriminant constraint of LDA.

D. Discrimenant Componenet Analysis (DCA)

The training data set is the same as described in Section II-C. Below shows the general steps of DCA:

1–2) The same as LDA's 1–2)

3) Compute the regulated between-class scatter matrix $\boldsymbol{S'_B}$, the regulated within-class scatter matrix $\boldsymbol{S'_W}$, and the regulated total scatter matrix $\boldsymbol{\bar{S}'}$:

$$\boldsymbol{S'_B} = \boldsymbol{S_B} + \rho'\boldsymbol{I} \qquad \boldsymbol{S'_W} = \boldsymbol{S_W} + \rho\boldsymbol{I} \tag{8}$$

$$\boldsymbol{\bar{S}'} = \boldsymbol{S'_B} + \boldsymbol{S'_W} = \boldsymbol{\bar{S}} + (\rho + \rho')\boldsymbol{I} \tag{9}$$

where ρ' and ρ are ridge parameters, and $\boldsymbol{\bar{S}} = \boldsymbol{S_B}+\boldsymbol{S_W}$.

4) Compute the eigenvectors $\{\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots, \boldsymbol{e_m}\}$ and corresponding eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ of scatter matrix $\boldsymbol{S_W^{-1}}\boldsymbol{\bar{S}'}$ through spectral decomposition, e.g. eigen decomposition.

5) Sort the eigenvectors by non-increasing eigenvalues and choose d eigenvectors with the largest eigenvalues to form a projection matrix $\boldsymbol{W_{dca}} \in \mathbb{R}^{m\times d}$, where each column is an eigenvector.

6) Transform each sample onto the new subspace:

$$\boldsymbol{x'_i} = \boldsymbol{W^T_{dca}} \times \boldsymbol{x_i} \tag{10}$$

where $\boldsymbol{x_i} \in \mathbb{R}^m$, and $\boldsymbol{x'_i} \in \mathbb{R}^d$.

$\boldsymbol{W_{dca}}$ is the DCA projection matrix. Similar to LDA, DCA could at most reduce the m dimensionality to $k-1$. There are works [8] discussing about the influence of parameters ρ and ρ' on the performance of DCA. In this paper, we set $\rho = 0.02$ and $\rho' = 0.1$.

III. Privacy-Preserving Face Recognition

A. Problem Description

We consider a two-party problem, where the data user (e.g. authorities) has a centralized face database (e.g. suspects), and the data owner (e.g. smartphone users) owns face images for testing. The goal of privacy-preserving face recognition is to allow the data user to determine if a face from the data owner is contained in his database, without compromising the privacy of data owner. Our privacy preserving face recognition framework contains three steps: feature extraction, privacy-preserving dimensionality reduction and classification. Below describes the details of the methods we utilized for each step to test our framework. However, in practice, the specific methods utilized in each step could also be replaced by other corresponding (more advanced) methods.

B. Feature Extraction

In this step, we transform the face image into feature vector (FV). Two feature extraction methods are utilized, respectively.

1) Pixel Feature: Pixel feature is a vector of all pixel values of a grayscale image. For instance, a 100×100 grayscale face image has a FV of $10,000$ length.

2) Gabor Feature: Gabor feature is utilized for edge detection and texture representation of images. For a face image, a set of Gabor filters are applied, and the downsized magnitude results forms its Gabor feature. For instance, applying 40 Gabor filters on a 100×100 face image, and downsizing each reuslt to 30×30, results in a FV of 36,000 length.

C. Privacy-preserving Dimensionality Reduction

To preserve the privacy of testing data, PCA, LDA and DCA are utilized to transform FV to dimension reduced feature vector (DRFV), respectively. Below shows the three dimensionality reduction methods in detail.

PCA. Suppose the projection matrix is $\boldsymbol{W_{pca}}$. Each testing phase begins with selecting a parameter d. Then, project FV on the $m \times d$ matrix $\boldsymbol{W_{pca}}$ to get $DRFV$.

PCA-LDA. In our case, the dimension of FV of a image is usually much larger than the number of training data. For LDA, this would result in (a) the within class scatter matrix $\boldsymbol{S_w}$ becoming singular, and (b) the overfitting of the transformed data. To overcome this issue, a two step dimensionality reduction method applies. In the training, the m-dimensional training data is projected to a r-dimensional subspace using a $m \times r$ PCA projection matrix $\boldsymbol{W_{pca}}$, where $r < n-k$, n is the number of training data, and k is the number of unique classes. Then, the r-dimensional data is projected into a $k-1$ dimensional subspace using a $r \times (k-1)$ LDA projection matrix $\boldsymbol{W_{lda}}$. In the testing, each FV is projected on a $m \times (k-1)$ matrix $\boldsymbol{W_{pca}} \cdot \boldsymbol{W_{lda}}$ to get $DRFV$.

PCA-DCA. Comparing with LDA, adding the ridge parameters in DCA makes the within class scatter matrix $\boldsymbol{S_w}$ non-singular. However, applying two step dimensionality reduction could also relax the overfitting issue and improve the efficiency. Therefore, we apply PCA before DCA in the same way as described for LDA.

PCA is the common step for all three dimensionality reduction methods. As discussed in Section II-B, the number of Principal Components (PCs) in PCA, namely parameter d, has an impact on the balance between privacy and utility. For instance, in our problem, the data user can determine the range of acceptable number of PCs they want to keep for them to have good utility of the data. The data owner can choose to share their data at any of the dimensions in that range, or not at all. Therefore, we could use the number of PCA PCs as the privacy policy level. In this paper, we use the privacy policy level and the number of PCs interchangeably.

D. Classification

Support Vector Machine (SVM) [9] is utilized as our classification method. Each subject is trained multiple models, where each model subjects to a feature type, a dimensionality reduction method and one privacy policy level. Each model specifies a threshold of the probability $\theta \in [0, 1]$.

In the testing phase, given a $DRFV$ of a face image, each two-class SVM model outputs a binary result and a probability calculated from the SVM decision value. If the probability is larger than a model's θ, we consider that the testing face belongs to the corresponding subject. The testing face might be recognized as multiple subjects, we return the subject whose model outputs the highest probability as the final result. If the testing face is not recognized as any subject, we consider it is not in the database.

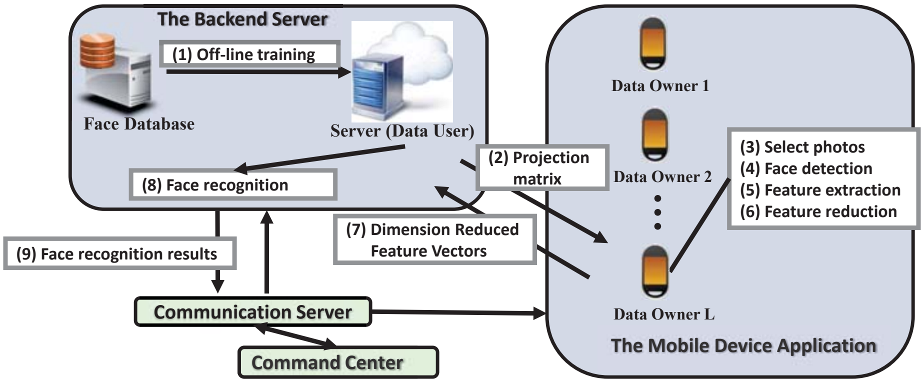

Fig. 1: FRiPAL Framework.

IV. FRiPAL System Design

FRiPAL is a privacy-preserving face recognition framework, which enables rapid photo collection and face recognition, while ensuring the data owner control over the data privacy. FRiPAL supports two feature extraction methods (Pixel Feature and Gabor Feature), and three dimensionality reduction methods (PCA, LDA and DCA). Fig. 1 shows the main components of FRiPAL: back-end server, mobile application, communication server, and command center.

Fig. 1 also shows the work flow of FRiPAL. The back-end server begins with an off-line training to prepare the classification models, mean vectors, projection matrices and data scale parameters. In each use case,

1) The mobile application updates the mean vectors and the projection matrices from the back-end server.
2) The end user selects photos for face detection. Then, the mobile application performs feature extraction and dimensionality reduction on the selected faces.
3) The mobile application sends DRFVs to the back-end server.
4) The back-end server performs face recognition and sends results to the command center.

Below describes the design and implementation details about each components.

A. Back-end server

This component provides the server side support of FRiPAL, including 1) information synchronization, 2) privacy-preserving face recognition, and 3) results update.

1) Information synchronization: This process synchronizes the mean vector and the projection matrix with the mobile application. In each use case, mobile application sends the client-side version number to the back-end server. The back-end server updates the newest projection matrix and mean vectors to the mobile application if any update is available.

2) Privacy-preserving face recognition: In each use case, the server receives DRFV, the feature type, the dimensionality reduction method, and privacy policy level from the mobile application. Then, each DRFV is tested against all the subjects' models with corresponding settings (Section III-D).

3) Results update: If a wanted subject (e.g. suspect) is recognized in a given photo, the back-end server sends the face recognition results to the command center.

B. mobile application

The mobile application is developed upon Android API level 23, including 1) face detection, 2) feature reduction, and 3) DRFV upload.

1) Face detection: The Haar Feature-based Cascade Classifiers [10] are adopted for face detection, which is implemented using OpenCV 3.0.0 library. The library contains a pre-trained classifier and API calls to do the face detection over an grayscale image.

2) Feature reduction: Given a detected grayscale face, the application first resizes it to a predefined width and height (e.g. 100×100). Then, the specified type of feature is extracted. The same dimensionality reduction procedure (Section III-C) applies on the FV to generate DRFV. As mentioned in Section III-C, the end user is allowed to specify the preferred privacy level (the number of PCs).

3) DRFV upload: The application uploads the DRFV, along with the feature type, the dimensionality reduction method and selected privacy policy level to the back-end server.

C. Communication Server and Command Center

Communication Server. The communication server is built upon RabbitMQ [11], which is an open source message broker software that supports the AMQP [12]. The mobile application works as a message producer, which creates and publishes messages to the communication server. The back-end server works as the message consumer that handles the message routed through the communication server.

Command Center. The command center displays the face recognition results and provides an interface for the agents and authorities to analyze and make further decision.

V. Experimental Evaluation

In this section, we evaluate FRiPAL through extensive experiments, in terms of accuracy, privacy and efficiency.

TABLE I: The Summaries of Experimental Datasets

	Number of Subjects (tr. / te.)	Number of Photos (tr. / te.)
Yale	28 / 28	5600 / 840
Gatech	0 / 50	0 / 714
Caltech	0 / 11	0 / 126
Total	28 / 89	5600 / 1680

A. *Experiment Setup*

1) Environment: The back-end server and commend center have been deployed on the same commodity computer with an 8 core Intel i7-4770 Processor, 32GB RAM, 400GB SSD, running 64-bit Ubuntu 14.04 LTS operating system. The communication server is a Ubuntu 14.04 LTS virtual machine, with 1 Processor, 8GB RAM and 20GB SSD, running on the same commodity computer. The mobile application has been deployed on two Android devices, a Nexus 5X and a Nexus 6P, respectively. The communication between the mobile devices and the commodity computer is through a wireless access point (MWR102 USB Powered Travel Router).

2) Dataset: The experiment dataset consists of data from three public datasets: the Caltech Faces 1999 (Caltech) [13], the Gatech Face Database (Gatech) [14], and the Yale Face Database B (Yale) [15]. More details for training data (tr.) and testing data (te.) are shown in Table I. The training data contains 28 subjects all from Yale, each has 200 photos. Thus, 5600 photos in total.

In order to make the experimental evaluation as unbiased and practical as possible, we generate the testing data from three different public datasets, which contains subjects both from and distinct from the training data. Specifically, the testing data consists of 28 subjects from Yale, each contains 30 photos; 50 subjects from Gatech, 714 photos in total; and 11 subjects from Caltech, 126 photos in total. Thus, 1680 photos in total.

3) Off-line Training: As discussed in Section III, we trained multiple models for each subject. Each training model subjects to a feature type, a dimensionality reduction method and one privacy policy level. In this experiment, we utilized two feature types, Pixel Feature and Gabor Feature and three dimensionality reduction methods, PCA, LDA and DCA, for the training of six types of models, namely, P-PCA, P-LDA, P-DCA, G-PCA, G-LDA and G-DCA. Furthermore, for each dimensionality reduction method, 18 different privacy policy levels (the number of PCA PCs) are selected, namely, 200, 190, 180, 170, 160, 150, 140, 130, 120, 110, 100, 90, 80, 70, 60, 50, 40, 30. For each subject in the training data, we utilize 200 (all) of his/her photos as the positive samples and 200 randomly selected photos of other subjects as the negative samples. For each subject, $2 \times 3 \times 18$ two-class SVM models are trained in total.

B. *Accuracy*

Threshold Selection. We use the ROC curve to select the threshold of each model (Section III-D), which is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. In our experiments, we use 101 threshold settings, namely, $\{0, 0.01, 0.02, \dots, 0.99, 1\}$. Fig. 2 illustrates six models of subject yaleB11 in Yale, where the x-axis is FPR and the y-axis is TPR. For each model, we consider the point of intersection of the ROC curves and $y = 1 - x$ as the threshold. For instance, in Fig. 2, the intersection point of P-PCA is around $(0.005, 0.91)$, and the corresponding threshold is 0.71, which means when the threshold is 0.71, the TPR is around 0.91 and the FPR is around 0.005.

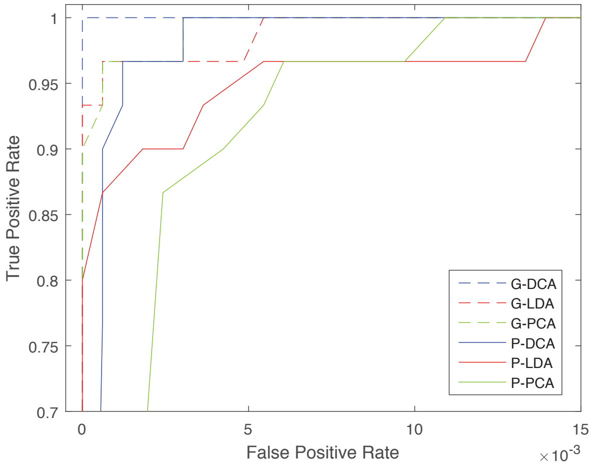

Fig. 2: The ROC curves of six models (P-PCA, P-LDA, P-DCA, G-PCA, G-LDA and G-DCA, with 200 PCA PCs) of subject yaleB11 in Yale.

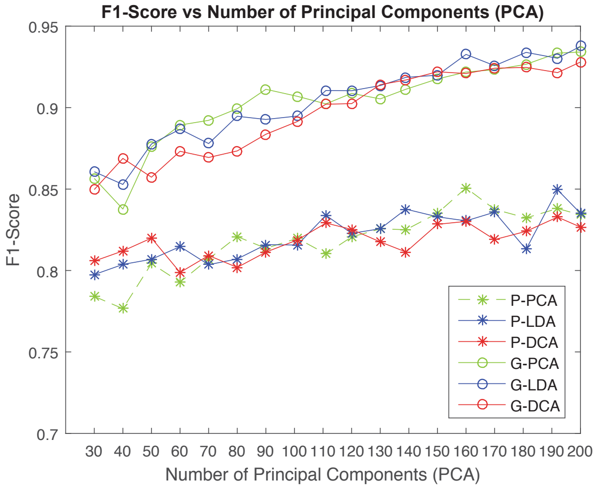

Fig. 3: F1-Score.

Accuracy Evaluation. We use F1-Score as the measure of accuracy. Fig. 3 shows the accuracy results of six methods. The x-axis shows 18 privacy policy levels. The y-axis shows the corresponding F1-Score results. Each result is the average over the results of all (28) subjects' models. It is clear that for three dimensionality reduction methods, as the number of PCs increasing, the accuracy increases gradually. It should be noted that the fluctuation in F1-Score is mostly due to the SVM parameter selection. The Gabor feature achieves an overall higher accuracy than the Pixel feature. The lowest accuracy is around 78% when we adopt P-PCA.

Considering the results mentioned above, for PCA, LDA and DCA, the pattern that the accuracy is positive correlated to the number of PCA PCs, is consistent.

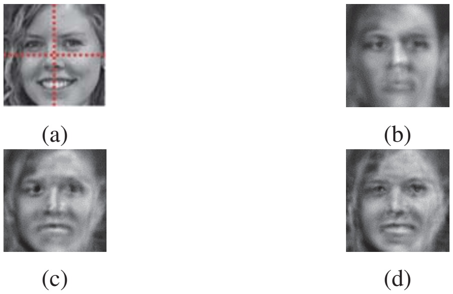

Fig. 4: PCA reconstruction of subject s10 in Gatech Face Database. (a) Original image. (b) 40 PCs reconstructed image (0.26 HS). (c) 120 PCs reconstructed image (0.35 HS). (d) 200 PCs reconstructed image (0.39 HS).

C. Privacy

Privacy Metrics. We utilized two metrics as the measure of privacy, namely Relative Error (RE) and Histogram Similarity (HS). The privacy metrics is conducted on the feature domain. We measure the difference between the original and the reconstructed FV, rather than the original image and reconstructed image. However, it is worth noting that the Pixel feature actually equals to the original image.

The RE is defined as equation (11), where N is the number of testing samples and m is the number of features. x_{ij} is the original value of jth feature in ith testing sample, and $\tilde{x}_{ij}$ is the corresponding value of the reconstructed data. Higher RE means more difference, thus, in our setting, more privacy is protected.

$$RE = \frac{1}{N \times m} \sum_{i=1}^{N} \sum_{j=1}^{m} |\frac{x_{ij} - \tilde{x}_{ij}}{x_{ij}}| \tag{11}$$

We utilize HS as the measure of image data's privacy. Since we only use grayscale images, the domain of feature value is $[0, 255]$, which is suitable for generating histograms. Let H and $\tilde{H}$ be the histograms of the original data and reconstructed data respectively. Then, the HS between H and $\tilde{H}$ is defined as equation (12), where M is the color dictionary (256) of grayscale images, and $S = \max(h_i - \tilde{h}_i)$, $i = 0, 1, \ldots, M-1$. The value of HS is in $[0, 1]$. Lower HS means less similarity, thus, in our setting, more privacy is protected.

$$HS(H, \tilde{H}) = \frac{1}{M} \sum_{i=1}^{M} (1 - \frac{|h_i - \tilde{h}_i|}{S}) \tag{12}$$

The histogram of an image describes its color distribution. Two distinct images may have a similar total color distribution, but it is rare that they have all the same partial color distributions. Therefore, we divide the image into grids. For instance, Fig. 4a is divided into four grids by the red dot lines. Then, we apply HS on each grid of the original data and the corresponding grid of the reconstructed data. Finally, we calculate the average HS among all the grids. In this experiments, we divide each image into 40 grids.

To demonstrate the effectiveness of HS, Fig. 4a is the original face, Fig. 4b is the reconstructed face after reduced the FV to 40 dimensions with PCA. Fig. 4d is the reconstructed face after reduced the FV to 200 dimensions with PCA. It can be seen that Fig. 4b is more blurred than Fig. 4d, while the HS value of Fig. 4b, 0.26 is lower than Fig. 4d, 0.39. It implies Fig. 4b preserves more privacy Fig. 4d. Therefore, the HS could be utilized to measure the privacy-preserving of images.

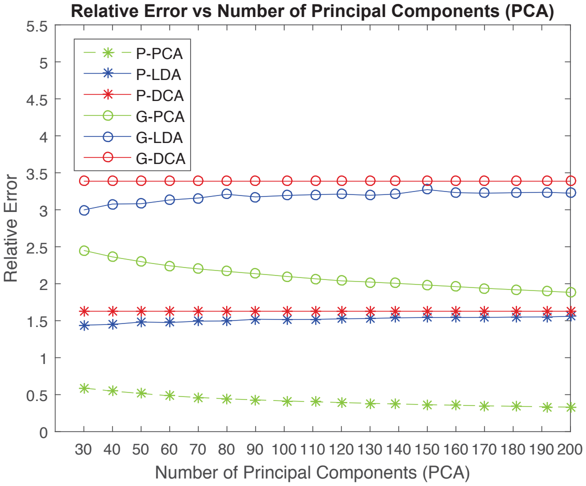

Fig. 5: Relative Error.

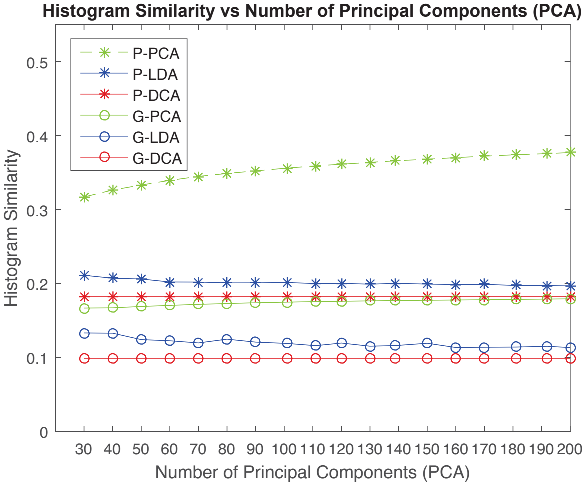

Fig. 6: Histogram Similarity.

Privacy Evaluation. Fig. 5 shows the RE of PCA, LDA and DCA, with two feature types. For all privacy policy levels listed, the RE of DCA is always the largest one, the RE of

PCA is always the smallest one, and the RE of LDA is always slight less than DCA. This pattern implies that DCA and LDA are more effective than PCA in terms of privacy preserving. Furthermore, as the privacy policy level increasing, the RE of PCA decreases, while the RE of DCA stays around the same value, and the RE of LDA slightly increases. This pattern shows that DCA and LDA are more consistent than PCA in terms of privacy preserving.

Fig. 6 shows the HS results, which presents the same patterns as the RE results. In addition, the maximum HS of LDA and DCA are less than the minimum HS of P-PCA (30 PCs). As shown in Fig. 4b, when using 40 PCs in PCA, the reconstructed face is already too blurred to be recognized from the original face. Therefore, LDA or DCA might get even more blurred reconstructed faces comparing with Fig. 4b.

Considering the results mentioned above, in terms of privacy preserving, DCA and LDA performs better and more consistent than PCA. From Fig. 3, it could be seen that PCA, DCA and LDA has similarity accuracy results, with the same feature types. Therefore, the data owners could choose to use DCA and LDA, which gives better and more consistent privacy protection, and could use the number of PCs to control the utility.

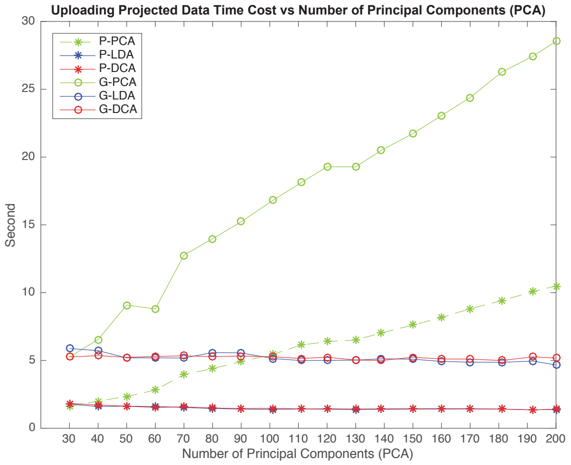

Fig. 7: Performance evaluation results on Nexus 6P.

D. Efficiency

We designed three experiments to show the system performance. The first experiment measures the time cost of updating projection matrix from the back-end server to the mobile application. The second experiment measures the time cost of the privacy preserving face recognition process, which is the core task of our proposed system. This process covers the face detection and feature reduction on the mobile device, the face recognition on the server side, and the data transmission between the two parties. The third experiment measures the time cost of uploading image to the back-end server. We select 10 images of each subject from the Yale testing data, which results in 280 images in total, and put them on the mobile device. We measure the performance with different privacy policy levels on P-PCA, P-LDA, P-DCA, G-PCA, G-LDA and G-DCA, respectively. In the second and third experiment, 10 images is grouped and processed together.

TABLE II: The Size of Different Projection Matrices

	Projection Matrix (privacy policy level)				
	40	80	120	160	200
P-PCA	3.2 MB	6.4 MB	9.6 MB	12.8 MB	16 MB
P-LDA	2.2 MB	2.2 MB	2.2 MB	2.2 MB	2.2 MB
P-DCA	2.2 MB	2.2 MB	2.2 MB	2.2 MB	2.2 MB
G-PCA	11.5 MB	23.0 MB	34.6 MB	46.1 MB	57.6 MB
G-LDA	7.8 MB	7.8 MB	7.8 MB	7.8 MB	7.8 MB
G-DCA	7.8 MB	7.8 MB	7.8 MB	7.8 MB	7.8 MB

Table II shows the size of different projection matrices. Table III summarizes the results of the first and second experiments. For instance, for G-PCA, with 200 PCs, when running on Nexus 6P, it takes 27 seconds to update the corresponding projection matrix (the size is 57.6 MB), and it takes 28.5 seconds to accomplish the privacy preserving face recognition of 10 images. Figure 7 shows the performance of the second experiment on Nexus 6P. It can be seen that, since DCA and LDA reduced the feature vector to an identical number, their performance are invariant against the change of privacy policy levels. For the PCA method, the more PCs, the more time it takes for processing. The experiment on Nexus 5X gives a similar result.

Considering all the experiment results above, FRiPAL is efficient, and through DCA and LDA, FRiPAL could preserve the privacy of data owners while maintaining the utility for data users.

VI. Related Works

Data Transformation for privacy-preserving. Data perturbation is an important technique for protecting the data privacy. [16] proposes to perturb the individual data with additive or multiplicative noise that is generated from certain distributions (e.g., Gaussian). [17] proposes to transforms the original whole data set by applying a random rotation matrix. However, both approaches suffer from a decreasing of the accuracy. Moreover, the first approach cannot resistant against the attack of noise filtering out [18], while the perturbed data obtained by the second approach can be restored by another rotation matrix [19]. Our proposed method maintains the utility, while is more resistant to the reconstruction attack.

Privacy-preserving face recognition. [20] has proposed the first privacy preserving face recognition. They consider a two party problem, data user owns a database of face images, and data owner wants to know whether the face image he owns is in data user's database. The data owner does not want to reveal the image nor the recognition result, while the data user does not want to leak the privacy of his face image database. To resolve the problem, an additively homomorphic public key encryption scheme has been used to securely calculate the distance between the data owner's data and the data in the database. Their work suffers from a

TABLE III: The Performance of UpdateProjectionMatrix and UploadProjectedData

	UpdateProjectionMatrix (second)				UploadProjectedData (second / 10 images)			
	Nexus 5X		Nexus 6P		Nexus 5X		Nexus 6P	
	30	200	30	200	30	200	30	200
P-PCA	1.305	10.647	2.058	7.566	1.331	10.797	1.608	10.493
P-DCA	1.325	1.294	1.020	1.208	1.634	1.197	1.833	1.41
P-LDA	1.266	1.277	1.133	1.024	1.472	1.212	1.786	1.378
G-PCA	3.878	30.881	3.924	27.062	6.251	29.957	5.248	28.543
G-DCA	5.135	4.476	4.030	3.653	7.198	5.365	5.259	5.170
G-LDA	4.344	4.116	3.499	3.861	6.982	5.2	5.88	4.667

heavy computation and communication cost by involving the homomorphic encryption, and it cannot be applied to other machine learning methods directly. [21] has proposed a hybrid solution using homomorphic encryption and garbled circuits, which improves the previous work by shifting most of the computation and communication cost to the pre-computation phase. Rather than encryption, we propose a more efficient and general method by using the dimensionality reduction methods.

VII. Conclusion

This paper explores the usage of dimensionality reduction techniques on privacy preserving face recognition. To demonstrate the proposed approach, we implement an efficient privacy preserving face recognition client server system, FRiPAL, using three dimensionality reduction methods, PCA, LDA and DCA with two types of features. The system performance is evaluated on two Android devices, Nexus 5X and Nexus 6P. The results confirm the efficiency of your system for real life usage. Through the extensive experiments, all three methods have similar accuracy results when using the same feature type. RE and HS is utilized to illustrate the privacy preserving performance. As the privacy policy level increasing, the privacy preserving of PCA degrades, while DCA and LDA has a more consistent and better results than PCA. Therefore, DCA and LDA maintain the utility while privade abetter privacy preserving. In the future, we will work on applying our methods on the multiple data owner and multiple data user scenario.

Acknowledgment

This material is based on research sponsored by the DARPA Brandeis Program under agreement number N66001-15-C-4068.[1]

References

[1] D. M. M. Lepinski, D. Levin and R. Watro, “Privacy-enhanced android for smart cities applications,” in *EAI International Conference on Smart Urban Mobility Services*, 2015.

[2] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[4] S. Laur, H. Lipmaa, and T. Mielikäinen, “Cryptographically private support vector machines,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 618–624.

[5] J. Vaidya and C. Clifton, “Secure set intersection cardinality with application to association rule mining,” *Journal of Computer Security*, vol. 13, no. 4, pp. 593–622, 2005.

[6] K. Liu, H. Kargupta, and J. Ryan, “Random projection-based multiplicative data perturbation for privacy preserving distributed data mining,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 1, pp. 92–106, 2006.

[7] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[8] S.-Y. Kung, “Discriminant component analysis for privacy protection and visualization of big data,” *Multimedia Tools and Applications*, pp. 1–36, 2015.

[9] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[10] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–511.

[11] “Rabbitmq - rabbit message queue,” Mar. 2016. [Online]. Available: https://www.rabbitmq.com/

[12] “Amqp: Advanced message queuing protocol,” Mar. 2016. [Online]. Available: https://www.amqp.org/

[13] “California institute of technology. faces 1999 (front),” Mar. 2016. [Online]. Available: http://www.vision.caltech.edu/html-files/archive.html

[14] A. Nefian and M. H. Hayes III, “A hidden markov model-based approach for face detection and recognition,” Ph.D. dissertation, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1999.

[15] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 643–660, 2001.

[16] R. Agrawal and R. Srikant, “Privacy-preserving data mining,” in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 439–450.

[17] K. Chen and L. Liu, “Privacy preserving data classification with rotation perturbation,” in *Data Mining, Fifth IEEE International Conference on*. IEEE, 2005, pp. 4–pp.

[18] Z. Huang, W. Du, and B. Chen, “Deriving private information from randomized data,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 37–48.

[19] K. Liu, C. Giannella, and H. Kargupta, “An attackers view of distance preserving maps for privacy preserving data mining,” in *Knowledge Discovery in Databases: PKDD 2006*. Springer, 2006, pp. 297–308.

[20] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, “Privacy-preserving face recognition,” in *Privacy Enhancing Technologies*. Springer, 2009, pp. 235–253.

[21] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, “Efficient privacy-preserving face recognition,” in *Information, Security and Cryptology–ICISC 2009*. Springer, 2009, pp. 229–244.

[1] The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.