Control of Photo Sharing over Online Social Networks

Kaihe Xu*, Yuanxiong Guo*, Linke Guo†, Yuguang Fang*, Xiaolin Li*
*Department of Electrical and Computer Engineering,
University of Florida, Gainesville, FL 32611, USA
{xukaihe, guoyuanxiong}@ufl.edu, {fang, andyli}@ece.ufl.edu
†Department of Electrical and Computer Engineering,
Binghamton University, Binghamton, NY 13902, USA
lguo@binghamton.edu

***Abstract*—Photo sharing is an attractive feature which popularizes Online Social Networks (OSNs). Unfortunately, it may leak users' privacy if they are allowed to post, comment, and tag a photo freely. In this paper, we attempt to address this issue and study the scenario when a user shares a photo containing individuals other than himself/herself (termed co-photo for short). To prevent possible leakage of a photo privacy, we design a mechanism to enable each individual in a photo be aware of the posting activity and participate in the decision making on the photo posting. For this purpose, we need an efficient facial recognition (FR) system that can recognize everyone in the photo. However, more demanding privacy setting may limit the number of the photos publicly available to train the FR system. To deal with this dilemma, our mechanism attempts to utilize users' private photos to design a personalized FR system specifically trained to differentiate possible photo co-owners without leaking his/her privacy. We have also developed a distributed consensus-based method to not only reduce the computational complexity, but also preserve the privacy during the training. We show that our system is superior to other possible approaches in terms of recognition ratio and efficiency. Our mechanism is implemented as an Android application on Facebook's platform.**

I. Introduction

OSNs have become integral part of our daily life and has profoundly changed the way we interact with each other, fulfilling our social needs–the needs for social interactions, information sharing, appreciation and respect. It is also this very nature of social media that makes people put more content, including photos, over OSNs without too much thought on the content. However, once something, such as a photo, is posted online, it becomes a permanent record, which may be used for purposes we may not expect. For example, a posted photo in a party may reveal a connection of a celebrity to a mafia world. Just because OSN users may be careless in posting content while the effect is so far-reaching, privacy protection over OSNs becomes an important issue. When more functions such as photo sharing and tagging are added, the situation becomes more complicated. When someone attempts to share a co-photo that contains individuals (photo co-owners) other than himself/herself, currently there is no restriction on posting. On the contrary, social network service providers like Facebook are encouraging users to post co-photos and tag their friends out in order to get more people involved. However, what if the co-owners of a photo are not willing to share the co-photo? Is it a privacy violation to share co-photos without permissions of the co-owners? Should the co-owners have some control over the co-photos?

With these questions, we shall take a new look at the popular OSNs such as Facebook to find how users' privacy are managed especially with the photo sharing feature. According to the statistics from Facebook, 95% of the users are tagged at least once on a photo, while most photos are tagged by someone else. Tagging reveals identity immediately, but it also works as a notification. There are 350 million photos uploaded everyday, and so what if someone just uploads photos containing a user without his/her knowledge? Currently, we can barely do anything about it.

In this paper, we propose a mechanism to help photo co-owners to get some control over their co-photos. We argue that the co-owner of a photo should have the same control over the photo as the owner. Whether or not to post the photo should be a collaborative decision of everybody in the photo. To do that, we need to design an add-on scheme to monitor photo posting activities on OSNs. Whenever a user attempts to post a photo, he/she will receive a notification and has to make the joint decision on whether to post it or not with all individuals involved in the photo. Thus, a face recognition engine (FR) is needed to recognize users in the photo. Photos to be posted usually contains social friends on OSN, and thus, FR can be trained to recognize the social friends (people in the social circle). Training techniques could be adapted from the off-the-shelf FR training algorithms. However, how to get enough training samples from the social circle in the OSN for the FR engine is the key. FR with higher recognition ratio demands more training samples (photos of friends). However, online photo resources may not contain enough photos of a friend potentially in a photo. Users who really care about their photo privacy are unlikely to post too many photos online. Perhaps it is exactly those people who really want to use our proposed mechanism and expect a high recognition ratio. To break this dilemma, we propose a privacy-preserving distributed collaborative training system for our FR with users' private photos as the training input.

Since the decision on photo posting involves friends in the circle (the most likely scenario) and it is distributed in nature, our problem can be transformed to be a typical secure multi-party computation problem. Intuitively, we may apply cryp-

This work was partially supported by National Science Foundation under grant CNS-1343356. The work of X. Li was partially supported by CCF-1128805 and ACI-1229576.

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

tographic technique to protect privacy, but the computational and communication cost may pose a serious problem for a large OSN. Besides cryptographic approach, we may take the consensus-based approach as a promising alternative. The idea is to let each user only deal with his/her local data and get the local training result, and then the neighboring users only need to exchange their training results. In the next round, each user still works on his own training data, but take the training results from his/her neighbors as references. We expect that the information will be spread over the OSN and everyone involved will reach the same conclusion. We will demonstrate that by using our method, we could make our FR mechanism distributed and computed in parallel, and thus, the privacy and efficiency can be achieved simultaneously.

The rest of this paper is organized as follows. In Section II, we review the related works. Section III presents the formulation of our problem and the assumptions in our study. In Section IV, we give a detailed description of the proposed mechanism, followed by Section V, conducting security analysis of the proposed mechanism. In Section VI, we describe our implementation on Android platform with the Facebook SDK and the extensive experiments to validate the accuracy and efficiency of our system. Finally, Section VII concludes the paper.

II. Related work

In [9], [10], Stone et al., for the first time, propose to use the contextual information in the social realm and co-photo relationship to do automatic FR. They define a pairwise conditional random field (CRF) model to find the optimal joint labeling by maximizing the conditional density. Specifically, they use the existing labeled photos as the training samples and combine the photo co-occurrence statistics and baseline FR score to improve the accuracy of face annotation. In [3], Choi et al. discuss the difference between the traditional FR system and the FR system that is designed specifically for OSNs. They point out that a customized FR system for each user is expected to be much more accurate in his/her own photo collections.

While intensive research interests lie in FR engines refined by social connections, the security and privacy issues in OSNs also emerge as important and crucial research topics. In [7], the privacy leakage caused by the poor access control of shared data in Web 2.0 is well studied. To deal with this issue, access control schemes are proposed in [5] and [2]. In these works, flexible access control schemes based on social contexts are investigated. However, in current OSNs, when posting a photo, a user is not required to ask for permissions of other users appearing in the photo. In [1], Besmer and Lipford study the privacy concerns on photo sharing and tagging features on Facebook. A survey [1] is used to study the effectiveness of the existing countermeasure of untagging and shows that users are worrying about offending their friends when untagging themselves. They also provide a tool to enable users to restrict others from seeing their photos when the photo is posted as a complement strategy to protect privacy. In [8], Squicciarini et al. propose a game-theoretic scheme in which the privacy policies are collaboratively enforced over the shard data. However, the ownership over the shard data cannot be determined automatically, and the potential owners of a shared data item are identified using the tagging features on the current OSNs.

Comparing with previous works, our contributions are as follows.

1) In our paper, the potential owners of the shared items (photos) can be automatically identified with/without user-generated tags.
2) We propose to use private photos in a privacy-preserving manner and social contexts to derive a personal FR engine for any particular user.
3) Orthogonal to the traditional cryptographic solution, we propose a consensus-based method to protect the local data (photos) while improving the efficiency.

III. Problem statement and hypotheses

We assume that $user_i$ has a photo set of size N_i of himself as his private training samples (say, stored on his/her own device such as smart phone). From the private photo set, a user detects and extracts the faces on each photo with the standard face detection method [12]. For each face, a vector of size p is extracted as the feature vector. Then, for the user i, his/her training set could be written as x_i of size $N_i \times p$. With social contexts, each user has a personal FR in which the potential photo posting users are restricted to a small group consisting of his/her friends and himself. Typically, this multi-class classification system is designed by combining several binary classifiers together with either one-vs-one method or one-vs-all method. However, no matter which method we use, it requires a centralized node to access all the training samples from each class. Therefore, in our scenario, it is required that the training samples (photos) are fed to the classifier in a privacy-preserving manner.

A FR engine for a large-scale social network may require discriminating millions of individuals. It seems to be a daunting task that can never be accomplished. However, we may be able to decompose it into several personal FR engines. Social contexts imply some information useful: if someone attempts to post a photo, it is very likely to be a photo of himself/herself or his/her friends[9]. In other words, the personal FR on an OSN does not need to worry about the strangers. We assume that for a user, say, i, the suspects on his/her photos are restricted to a small group of himself and his one-hop neighbors (e.g., friends), denoted as the neighborhood $\mathcal{B}_i$. Then our goal for the personal FR at user i is to differentiate users in $\mathcal{B}_i$. For a test photo x of the user i, all the faces in it are identified as a set of users $\mathcal{I}$. Request for permission along with x is send to the users in $\mathcal{I}$ except for i.

IV. System Overview

In this section, we present the detailed description of our system. First we use a toy system with two users to demonstrate the principle of our design. Then, we discuss how to build a general personal FR with more than two parties.

A. A toy system

Suppose there are only two users $user_1$ and $user_2$ with private training data x_1 and x_2. In order to distinguish them, we only need to find a binary decision function $f(\cdot)$. When a

probing sample x comes, if $f(x) > 0$, x belongs to $user_1$ and vice versa. In this paper, the decision function is determined by the support vector machine as $f(x) = K(w, x) + b$, where $K(\cdot, \cdot)$ is the kernel function. For the ease of description, here we use a linear kernel. For the training samples x_i of size $N_i \times p$, where N_i is the number of training samples, and p is the number of features in each training sample. Denote u as $u = [w, b]$ of size $(p+1) \times 1$, X_i as $X_i = [x_i, 1]$ of size $N_i \times (p+1)$ and Y_i is a $N_i \times N_i$ diagonal matrix indicating the class labels of samples in X_i. If X_1 is the positive sample set and X_2 is the negative sample set, the corresponding label matrix Y_1 is an identity matrix and Y_2 is a negative identity matrix. Meanwhile, a diagonal matrix Π is introduced to guarantee that $\frac{1}{2}u^T\Pi u = \frac{1}{2}w^T w$. Then Π is constructed as a $(p+1) \times (p+1)$ diagonal matrix with $\Pi(i, i) = 1$ for $i = 1, 2, \ldots, p$ and $\Pi(p+1, p+1) = 0$. In this case, the decision function $f(\cdot)$ can be obtained by solving the following problem:

$$\begin{aligned} \min_{u, \xi_1 \geq 0, \xi_2 \geq 0} \quad & \frac{1}{2}u^T\Pi u + C|\xi_1| + C|\xi_2| \\ \text{s.t.} \quad & Y_1 X_1 u \geq 1 - \xi_1, \\ & Y_2 X_2 u \geq 1 - \xi_2. \end{aligned} \tag{1}$$

In problem (1), by minimizing $\frac{1}{2}u^T\Pi u$, we find u such that the distance between the nearest data points from X_1 and X_2 to u is maximized. The first and second constraints in problem (1) is used to ensure that the decision function $f(x) = X_{1k}u > 1$ for $k = 1...N_1$ and $f(x) = X_{2k}u < -1$ for $k = 1...N_2$. ξ_i is a set of slack variables in case the training samples are not separable. If a certain positive sample X_{1k} cannot make $X_{1k}u > 1$, a positive slack variable ξ_{1k} is assigned so that $X_{1p}u > 1 - \xi_{1p}$. Meanwhile, a penalty of $C\xi_{1p}$ is also assigned to the objective function, where C is the user-chosen penalty parameter and vice versa for the negative samples. Notice that for each user, his/her private training set corresponds to a set of private constraints. An alternative approach other than cryptography should be consensus-based method. The idea is that: at the iteration of t, $user_1$ makes a reasonable guess of u as u_1^t and sends u_1^t to $user_2$. Upon receiving u_1^t, $user_2$ uses it as a reference for the estimation of u_2^{t+1}. To understand this estimation refinement, first we need to reformulate problem (1) in its equivalent form that could easily be distributed as in problem (2), which can be solved using the alternating directional method of multipliers (ADMM) based consensus algorithm.

$$\begin{aligned} \min_{\{u_i, \xi_i \geq 0\}} \quad & \sum_{i=1,2} \frac{1}{4}u_i^T\Pi u_i + C\sum_{i=1,2}|\xi_i| \\ \text{s.t.} \quad & Y_i X_i u_i \geq 1 - \xi_i, && i = 1, 2 \\ & u_i = u_j, && i, j = 1, 2. \end{aligned} \tag{2}$$

Problem (2) can be solved through its dual problem. For this purpose, the augmented Lagrange function with the Language multipliers of $\{\lambda_i\}$ and $\{\alpha_i\}$ can be written as:

$$\begin{aligned} \mathcal{L}(\{u_i\}, \{\lambda_i\}, \{\alpha_i\}) = & \frac{1}{4}\sum_{i=1,2} u_i^T\Pi u_i + \sum_{i,j=1,2} \alpha_i^T(u_i - u_j) \\ & - \sum_{i=1,2} \lambda_i^T(Y_i X_i u_i - 1 + \xi_i) + \sum_{i,j=1,2} \frac{\rho}{2}\|u_i - u_j\|^2. \end{aligned} \tag{3}$$

In Eq. (3), we omit the Language multipliers of the slack variables, which can be canceled out in the Wolfe dual problem. Here, $\frac{\rho}{2}\|u_i - u_j\|^2$ is the regularization term, which has two roles: (1) It eliminates the condition that $\mathcal{L}$ is differentiable such that the solution converges under far more general conditions. (2) By adjusting the parameter of ρ, we can trade off the speed of convergence for better steady-state approximation[4].

In Eq. (3), $\mathcal{L}$ can be minimized in a cyclic fashion with the method of multipliers: at each iteration, $\mathcal{L}$ is minimized with respect to one variable while keeping all other variables fixed. Meanwhile, the multipliers $\{\alpha_i\}$ should also be updated with the equality constraint residual. The method of multipliers to update the variables at each iteration $t+1$ is summarized as follows,

$$\begin{aligned} \{u_i^{t+1}\} = & \operatorname*{argmin}_{\{u_i\}} \mathcal{L}(\{u_i\}, \{\lambda_i^t\}, \{\alpha_i^t\}); \\ \alpha_i^{t+1} = & \alpha_i^t + \rho(u_i^{t+1} - u_j^{t+1}), \quad i \neq j. \end{aligned} \tag{4}$$

In (4), u_i is calculated with the KKT condition at each iteration. If the calculation of u_i^{t+1} is only related to X_i, Y_i, λ_i^t and u_j^t, but not to the private constraints of user j, we can assign this task to user i with the privacy of X_j preserved. By requesting u_j^t from user j and using his/her local data, u_i^{t+1} can be obtained locally at user i. In this way, the secure collaborative training can be achieved without using cryptographic tools (we will present the detailed security analysis of this method in Section V). In the following part of this section, we focus on the local update of u_i^t.

In general, the quadratic programming problem in the form of problem (2) is solved through its Wolfe dual problem. First, user i takes the partial derivatives of $\mathcal{L}$, get the expression of u_i^t with respect to λ_i^t and substitute it back in $\mathcal{L}$. With the duality property, u_i^t is obtained. The Wolfe dual and the solution of u_i^t is listed in Eq. (5).

$$\begin{aligned} \lambda_i^{t+1} = & \operatorname{argmax}\big\{-\lambda_i^T Y_i X_i(\Pi + 4\rho I)^{-1} X_i^T Y_i \lambda_i \\ & + [1 + 2Y_i X_i(\Pi + 4\rho I)^{-1}(\alpha_i^t - \alpha_j^t - 2\rho u_j^t)]^T \lambda_i\big\} \\ u_i^{t+1} = & 2(\Pi + 4\rho I)^{-1}[X_i^T Y_i \lambda_i^{t+1} - (\alpha_i^t - \alpha_j^t) + 2\rho u_j^t] \\ \alpha_i^{t+1} = & \alpha_i^t + \rho(u_i^{t+1} - u_j^{t+1}). \end{aligned} \tag{5}$$

Regarding our security statement with (5), λ_i^{t+1} and u_i^{t+1} are only related to X_i, α_i^t and Y_i, the required external parameters are u_j^t and α_j^t. Hence, at the beginning of the iteration $t+1$, users need to exchange their local update of u_i^t and α_i^t from the previous iteration. The iterative update process can be summarized in Algorithm 1. We denote $u_{ij} = F(X_i, X_j)$ as the computation of classifier u_{ij} with X_i as positive training samples and X_j as negative training samples, then the computation of u_{ij} is given below: In Algorithm 1, $\text{qd}(A, B)$ is a standard quadratic programming solver that gives the optimal solution of $\max\{-\frac{1}{2}x^T Ax + B^T x\}$, and notice that we omit the constraint of $0 \leq \lambda \leq C$ for brevity. $threshold$ is the user-defined stopping criteria, a larger $threshold$ results with less iterations while a larger discrepancy between u_i and u_j.

Algorithm 1: Iterative Method to Compute u_{ij}

Input: Positive samples X_i, Negative samples X_j
Output: The classifier $f_{ij}(x) = x \cdot u_{ij}$

initial $\lambda, u_i^t, u_j^t, \alpha_i^t, \alpha_j^t$ as vectors of all zeros;
$A = 2X_i(\Pi + 4\rho I)^{-1}X_i^T$;
for *t = 0, 1, 2...* **do**
  $B = 1 + 2X_i(\Pi + 4\rho I)^{-1}(\alpha_i^t - \alpha_j^t - 2\rho u_j^t)$;
  $\lambda^{t+1} = \text{qd}(A, B)$;
  $u_i^{t+1} = 2(\Pi + 4\rho I)^{-1}[X_i^T\lambda^{t+1} - (\alpha_i^t - \alpha_j^t) + 2\rho u_j^t]$;
  if $|u_i^{t+1} - u_i^t| <$*threshold* **then**
    break;
  else
    $\alpha_i^{t+1} = \alpha_i^t + \rho(u_i^{t+1} - u_j^{t+1})$;
    send u_i^{t+1} and α_i^{t+1} to user j;
    request u_j^{t+1} and α_j^{t+1} from user j;
  end
end
return u_i^{t+1};

B. OSNs with social contexts

In the previous section, we show how to achieve privacy-preserving collaborative training in a toy system with two users. When we consider the practical scenario, each user may have more than one friend. In this case, we need an efficient multi-class classifier. Generally speaking, a multi-class classifier is achieved by using one of the two strategies to combine several binary classifiers: one-against-all and one-against-one.

For the strategy of one-against-one, a binary classifier $f_{jk}(\cdot)$ is established between any two users in $\mathcal{B}_i$. For a test sample x, if $f_{jk}(x) > 0$, we say that x passes the test of user j and the vote for user j is added by one. Then, after testing all the $\frac{1}{2}D_i(D_i+1)$ classifiers, x is recognized as the user with the maximum vote. In this case, there exists $\frac{1}{2}D_i(D_i+1)$ classifiers in $\mathcal{B}_i$. For a certain node j in $\mathcal{B}_i$, he/she needs to participate in D_i classifiers. At the same time, node j should be involved in D_j neighborhoods. Therefore, for each node, the average computation overhead is $\mathcal{T}_o\,\mathrm{O}\big(n^\epsilon \bar{D}^2\big)$, where $\mathcal{T}_o$ is the average number of iterations to converge. Generally speaking, one user needs to establish classifiers between {himself/herself, his/her one-hop neighbors} and {himself/herself, his/her two-hop neighbors}. After the classifiers are established, a decision tree is constructed to classify the test photo[6].

To do so, we modify the DAGSVM method for the one-against-one strategy in [6] towards the construction of the decision tree. DAGSVM has a decision tree structure by arranging $\frac{1}{2}\bar{D}(\bar{D}+1)$ classifiers properly. With a test sample x, if x passes the classifier f_{jk}, then go left, otherwise, go right. Finally, after reaching a leaf node, we find the correct class of x. However, it is only designed to make the decision among the known classes. If we use the DAGSVM directly, there should be a strong assumption: *users in each other's photos are all friends*. In reality, this is not the case. For example, Bob has a co-photo with him and Alice at a popular attraction spot. It is very likely that a face from a stranger may also be captured in the photo. According to the DAGSVM, the random faces must be recognized as a certain one-hop neighbor. Suppose it is Tom, then, the photo is sent to Tom, who is not supposed to see the photo before it is allowed to be posted.

However, detection of strangers (or outliers) is a well-known difficult problem. A *stranger* to user i in our application is a user who is not connected to user i on the friendship graph. We cannot train a classifier to recognize that person because we even do not know where to request the training samples. In this paper, we attempt to find out the strangers based on a *contradictory decision* from the decision tree. We have the following observations: (1) If a certain class participates in the training process, then a probing sample from it will get the correct result by following the decision tree. (2) If a certain class does not participate in the training process, then the classifier will give an unpredictable decision for a probing sample from this class. Based on these two observations, we propose a improved DAGSVM decision tree to capture the possible *contradictory decisions* as the evidence to determine a stranger.

V. Security analysis

Due to the fact that our whole system is composed of multiple subsystems, the security analysis in this section is built on top of the analysis of the subsystems. For two users, Alice and Bob, in a subsystem, we assume that they are semi-honest, which means both Alice and Bob follow the protocol, but they are also curious in the sense that they will store all the exchanged data and try to deduce the training photo set from it. The analysis is done on behalf of Alice (Alice stores all the data and tries to find the private photo set of Bob X_b) and the analysis for Bob can be similarly done.

Assume there are $\mathcal{T}_o$ rounds of parameter exchanges, then for Alice, she knows $\{u_b^t, \alpha_b^t\}$, for $t = 1...\mathcal{T}_o$ and the parameters of her own $\{u_a^t, \alpha_a^t\}$, for $t = 1...\mathcal{T}_o$. The goal is to find X_b which is a $N_b \times (p+1)$ matrix with $N_b \times p$ unknowns. Alice is familiar with the FR mechanism and she knows that the parameters at hand have the relationship as follows:

$$A = 2X_b c^{-1} X_b^T, \tag{6}$$

$$X_b^T\lambda = cu_b^t + d, \tag{7}$$

$$B = 1 + X_b c^{-1} d, \tag{8}$$

$$\lambda = \arg\min_{0\le\lambda\le\frac{C}{2}} \quad \frac{1}{2}\lambda^T A\lambda + B^T\lambda \tag{9}$$

where $c = 2(\Pi+4\rho I)^{-1}$, $d = \alpha_b^t - \alpha_a^t - 2\rho u_a^t$ can be computed accordingly for each iteration. At each iteration, Alice records the value of $X_b^T\lambda$ and tries to find out X_B. Notice that the value of λ comes from the quadratic optimization problem (9), in which A is a fixed matrix determined by X_i, B is changing with iterations, and the solution u_b is changing accordingly. We need to show that, with multiple $\{B, u_b\}$ tuples, Alice cannot get any information of X_b. To solve the quadratic optimization problem 9, we need to first find its Lagrange function:

$$\mathcal{L} = \frac{1}{2}\lambda^T A\lambda + B^T\lambda + \tau^T(\lambda - \frac{C}{2}) - \nu^T\lambda, \tag{10}$$

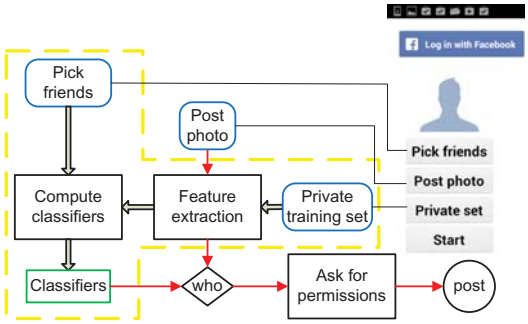


Fig. 1. System structure of our application

where τ and ν are Lagrange multipliers. The solution of problem (9) can be obtained through the KKT conditions:

$$-A\lambda - \tau + \nu = B^T \tag{11}$$

$$\tau^T(\lambda - c) = 0 \tag{12}$$

$$\nu^T x = 0 \tag{13}$$

$$0 \leq \lambda \leq \frac{C}{2}, \nu, \tau \geq 0.$$

With Eq.(6) and (7), Eq.(11) can be written as

$$-X_b(c \cdot u_b^t + d) - \tau + \nu = B. \tag{14}$$

If the parameters $\{c, u_b^t, d, B, \tau, \nu\}$ are known to Alice, she can get N_b equations, one for each training sample at Bob. With more than p iterations, she should be able to recover X_b by having enough equations to find out $N_b \times p$ unknowns. However, the Lagrange multipliers τ and ν are calculated when Bob is trying to solve problem (9) at each iteration and he will not reveal these parameters to Alice. τ and ν are easy to compute for Bob with matrix A, but for Alice, there is no clue to make a reasonable guess of them. In fact, the support vectors of Bob are determined by the combination of τ and ν. It is the support vectors of Alice and the support vectors of Bob that jointly define the hyperplane as u_{ab}. In this way, at each iteration, by revealing N_b equations, $2N_b$ unknowns are introduced. It is infeasible for Alice to have enough equations to find out X_b.

VI. Evaluation

In this section, we present the implementation of our algorithm on Andorid devices and test the facial recognition ratio over the database of "face Recognition Data, University of Essex, UK" to assign training set for each simulated users. The database contains photo for 395 individuals and 20 images per individual with varying poses and facial expressions. Each user is assigned with photos from the same individual randomly.

A. Implementation

Our prototype application is implemented on Google Nexus 7 tablets with Android 4.2 Jelly Bean (API level 17) and Facebook SDK. We use OpenCV Library 2.4.6 to carry out the face detection and Eigenface method to carry out the FR. Fig.1 shows the graphical user interface (GUI). The blue button located at the northeast is used for log in/out with Facebook. After logging in, a greeting message and the profile picture will be shown. Our prototype works in three modes: a setup mode, a sleeping mode and a working mode.

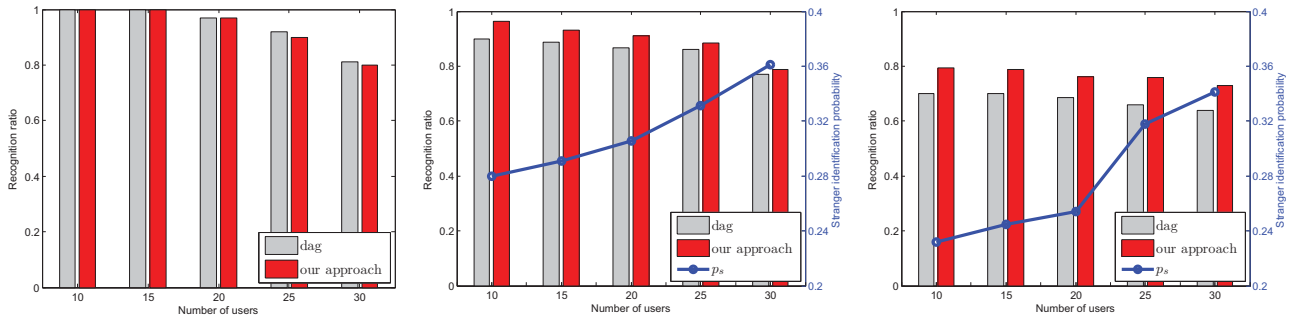
Running in the setup mode, the program is working towards the establishment of the improved decision tree. For this purpose, the private training set X_i and the neighborhood $\mathcal{B}_i$ need to be specified. X_i could be specified by the user with the button "Private training set". When it is pressed, photos in the smart phone galleries could be selected and added to X_i. To setup the neighborhood $\mathcal{B}_i$, at this stage, a user needs to manually select their Facebook friends with the button "Pick friends". Notice that, all the selected friends are required to install our application to carry out the collaborative training. With X_i and $\mathcal{B}_i$ specified, the setup mode could be activated by pressing the button "Start". Key operations and the data flow in this mode are enclosed by a yellow dashed box on the system architecture Fig.1. During the training process, a socket is established between any two users in $\mathcal{B}_i$ to enable the real-time communication. Through the sockets, Algorithm.1 is performed to obtain the classifiers. After the classifiers are obtained, an improved decision tree is constructed and the program switches from the setup mode to the sleeping mode.

The program could be invoked to the working mode with the button "Post Photo". Working in this mode, all the faces on the posting photo are detected and recognized, then, requests will be send to the detected friends. The requests will be shown as a notification in their App Center Requests. Along with the request, the photo will be sent to them. Upon receiving the post permissions from the detected friends, the photo is posted, otherwise, the posting action will be expired in two weeks. The data flow for a photo posting activity is illustrated by the solid red arrows. After the requests are sent out, the program will go back to the sleeping mode. If X_i or $\mathcal{B}_i$ is modified, the program will be invoked to the setup mode. In this case, the operations in the yellow dashed box will be performed again and the improved decision tree will be updated.

B. Facial recognition performance

In this subsection, we test the recognition accuracy of our design. We study the recognition ratio against the number of friends and the number of strangers. To detect a face on an image, we use the standard face detection in [12] and we use eigenface method [11] to extract features and vectorize the training image. However, the standard eigenface method is a centralized approach, it may not be applicable to our distributed case. For this reason, we use a universal eigenface to extract features for all the users in our network. Based on the simulation results, we find that this modification is reasonable due to the fact that the important features on human face lie on only a few directions which are the same to everyone. Facial feature extraction is beyond the scope of this paper. Better facial feature extraction method can be applied to our system to obtain a better recognition ratio.

In Fig.2, we show the recognition ratios of our proposed scheme and the scheme with DAG decision tree. As in Fig.2(a), when there are no strangers, both our proposed scheme and the DAG scheme could achieve very high recognition ratio of more than 80% when the number of users is fewer than 30. While in Fig.2(b), among the users, 10% of them are strangers, we can see that the recognition ratio of our



(a) No stranger (b) 10% strangers (c) 30% strangers

Fig. 2. Recognition ratio with varying number of users

scheme has a higher recognition ratio than the DAG scheme by 5%. The reason is that our scheme considers the stranger detection feature. The solid line represents the recognition ratio of the strangers p_s, which is increasing with the number of users. Intuitively, if there are more users, there will be more classifiers. As a result, the chance that a stranger gets the contradictory decision will be higher. Fig.2(c) shows a similar case where there are 30% strangers. In this case, our scheme is wining over the DAG scheme by 10% in terms of recognition ratio. This is achieved by the ability of identifying strangers. With 30 users, the probability of identifying a stranger is around 35%.

VII. Conclusions

Photo sharing is probably the most popular feature in online social networks such as Facebook. Unfortunately, careless photo posting may reveal privacy of individuals in a posted photo. To curb the privacy leakage, we propose to enable individuals potentially in a photo to give the permissions before posting. To do so, we need a privacy-preserving face recognition system to identify the images of individuals in the photo. In this paper, we develop a distributed consensus-based face recognition system to identify the images of individuals in a photo with high accuracy and low computation cost without leaking private information about those individuals. Our analysis shows the effectiveness and the efficiency of our proposed scheme. We expect that our proposed scheme will be very useful in protecting users' privacy in photo/image sharing over online social networks.

References

[1] A. Besmer and H. Richter Lipford. Moving beyond untagging: photo privacy in a tagged world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1563–1572, New York, NY, USA, 2010. ACM.

[2] B. Carminati, E. Ferrari, and A. Perego. Rule-based access control for social networks. In R. Meersman, Z. Tari, and P. Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, volume 4278 of *Lecture Notes in Computer Science*, pages 1734–1744. Springer Berlin Heidelberg, 2006.

[3] K. Choi, H. Byun, and K.-A. Toh. A collaborative face recognition framework on a social network platform. In *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pages 1–6, 2008.

[4] P. A. Forero, A. Cano, and G. B. Giannakis. Consensus-based distributed support vector machines. *J. Mach. Learn. Res.*, 99:1663–1707, August 2010.

[5] R. J. Michael Hart and A. Stent. More content - less control: Access control in the web 2.0. In *Proceedings of the Workshop on Web 2.0 Security and Privacy at the IEEE Symposium on Security and Privacy*, 2007.

[6] J. C. Platt, N. Cristianini, and J. Shawe-taylor. Large margin dags for multiclass classification. In *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

[7] D. Rosenblum. What anyone can know: The privacy risks of social networking sites. *Security Privacy, IEEE*, 5(3):40–49, 2007.

[8] A. C. Squicciarini, M. Shehab, and F. Paci. Collective privacy management in social networks. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 521–530, New York, NY, USA, 2009. ACM.

[9] Z. Stone, T. Zickler, and T. Darrell. Toward large-scale face recognition using social network context. *Proceedings of the IEEE*, 98(8):1408–1415.

[10] Z. Stone, T. Zickler, and T. Darrell. Autotagging facebook: Social network context improves photo annotation. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[11] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[12] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.