This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

My Privacy My Decision: Control of Photo Sharing on Online Social Networks

Kaihe Xu, *Student Member, IEEE,* Yuanxiong Guo, *Member, IEEE,* Linke Guo, *Member, IEEE,* Yuguang Fang, *Fellow, IEEE,* Xiaolin Li, *Member, IEEE*

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 privacy leakage of a photo, 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 their privacy. We also develop a distributed consensus-based method to reduce the computational complexity and protect the private training set. We show that our system is superior to other possible approaches in terms of recognition ratio and efficiency. Our mechanism is implemented as a proof of concept Android application on Facebook's platform.

Index Terms—Social network, photo privacy, secure multi-party computation, support vector machine, collaborative learning

1 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 never expect. For example, a posted photo in a party may reveal a connection of a celebrity to a mafia world. 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. For instance, nowadays we can share any photo as we like on OSNs, regardless of whether this photo contains other people (is a co-photo) or not. Currently there is no restriction with sharing of co-photos, on the contrary, social network service providers like Facebook are encouraging users to post co-photos and tag their friends in order to get more people involved. However, what if the co-owners of a photo are not willing to share this photo? Is it a privacy violation to share this co-photo without permission of the co-owners? Should the co-owners have some control over the co-photos?

To answer these questions, we need to elaborate on the privacy issues over OSNs. Traditionally, privacy is regarded as a state of social withdrawal. According to Altman's privacy regulation theory [1][15], privacy is a dialectic and dynamic boundary regulation process where privacy is not static but *"a selective control of access to the self or to ones group"*. In this theory, "dialectic" refers to the openness and closeness of self to others and "dynamic" means the desired privacy level changes with time according to environment. During the process of privacy regulation, we strive to match the achieved privacy level to the desired one. At the optimum privacy level, we can experience the desired confidence when we want to hide or enjoy the desired attention when we want to show. However, if the actual level of privacy is greater than the desired one, we will feel lonely or isolated; on the other hand, if the actual level of privacy is smaller than the desired one, we will feel over-exposed and vulnerable.

Unfortunately, on most current OSNs, users have no control over the information appearing outside their profile page. In [21], Thomas, Grier and Nicol examine how the lack of joint privacy control can inadvertently reveal sensitive information about a user. To mitigate this threat, they suggest Facebook's privacy model to be adapted to achieve multi-party privacy. Specifically, there should be a mutually acceptable privacy policy

- *K. Xu, Y. Fang, X. Li are with the Department of Electrical and Computer Engineering, Gainesville, FL 32611, USA. E-mail: xukaihe@ufl.edu, {fang, andyli}@ece.ufl.edu.*
- *L. Guo is with Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902, USA. E-mail: lguo@binghamton.edu.*
- *Y. Guo is with School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA. E-mail: richard.guo@okstate.edu.*

This work was supported in part by the U.S. National Science Foundation under Grants CNS-1343356 and CNS-1423165. The work of X. Li was partially supported by CCF-1128805 and ACI-1229576.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

determining which information should be posted and shared. To achieve this, OSN users are asked to specify a privacy policy and a exposure policy. Privacy policy is used to define group of users that are able to access a photo when being the owner, while exposure policy is used to define group of users that are able to access when being a co-owner. These two policies will together mutually specify how a co-photo could be accessed. However, before examining these policies, finding identities in co-photos is the first and probably the most import step. In the rest of this paper we will focus on a RF engine to find identities on a co-photo.

FR problems over OSNs are easier than a regular FR problem because the contextual information of OSN could be utilized for FR[20]. For example, people showing up together on a co-photo are very likely to be friends on OSNs, and thus, the FR engine could be trained to recognize social friends (people in social circle) specifically. Training techniques could be adapted from the off-the-shelf FR training algorithms, but how to get enough training samples is tricky. FR engine with higher recognition ratio demands more training samples (photos of each specific person), but online photo resources are often insufficient. Users care about privacy are unlikely to put photos online. Perhaps it is exactly those people who really want to have a photo privacy protection scheme. To break this dilemma, we propose a privacy-preserving distributed collaborative training system as our FR engine. In our system, we ask each of our users to establish a private photo set of their own. We use these private photos to build personal FR engines based on the specific social context and promise that during FR training, only the discriminating rules are revealed but nothing else.

With the training data (private photo sets) distributed among users, this problem could be formulated as a typical secure multi-party computation problem. Intuitively, we may apply cryptographic technique to protect the private photos, but the computational and communication cost may pose a serious problem for a large OSN. In this paper, we propose a novel consensus-based approach to achieve efficiency and privacy at the same time. The idea is to let each user only deal with his/her private photo set as the local train data and use it to learn out the local training result. After this, local training results are exchanged among users to form a global knowledge. In the next round, each user learns over his/hers local data again by taking the global knowledge as a reference. Finally the information will be spread over users and consensus could be reached. We show later that by performing local learning in parallel, efficiency and privacy could be achieved at the same time.

Comparing with previous works, our contributions are as follows.

1) In our paper, the potential owners of 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 achieve privacy and efficiency.

The rest of this paper is organized as follows. In Section 2, we review the related works. Section 3 presents the formulation of our problem and the assumptions in our study. In Section 4, we give a detailed description of the proposed mechanism, followed by Section 5, conducting performance analysis of the proposed mechanism. In Section 6, 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 7 concludes the paper.

2 Related Work

In [12], Mavridis et al. study the statistics of photo sharing on social networks and propose a three realms model: "a social realm, in which identities are entities, and friendship a relation; second, a visual sensory realm, of which faces are entities, and co-occurrence in images a relation; and third, a physical realm, in which bodies belong, with physical proximity being a relation." They show that any two realms are highly correlated. Given information in one realm, we can give a good estimation of the relationship of the other realm. In [19], [20], 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 [6], 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. A similar work is done in [5], in which Choi et al. propose to use multiple personal FR engines to work collaboratively to improve the recognition ratio. Specifically, they use the social context to select the suitable FR engines that contain the identity of the queried face image with high probability.

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 [17], 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 [13] and [4]. In these works, flexible access control schemes based on social contexts

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

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 [2], Besmer and Lipford study the privacy concerns on photo sharing and tagging features on Facebook. A survey was conducted in [2] to study the effectiveness of the existing countermeasure of untagging and shows that this countermeasure is far from satisfactory: users are worrying about offending their friends when untagging. As a result, they provide a tool to enable users to restrict others from seeing their photos when posted as a complementary strategy to protect privacy. However, this method will introduce a large number of manual tasks for end users. In [18], Squicciarini et al. propose a game-theoretic scheme in which the privacy policies are collaboratively enforced over the shared data. Each user is able to define his/her privacy policy and exposure policy. Only when a photo is processed with owner's privacy policy and co-owner's exposure policy could it be posted. However, the co-owners of a co-photo cannot be determined automatically, instead, potential co-owners could only be identified by using the tagging features on the current OSNs.

3 Problem Statement and Hypotheses

3.1 Privacy policy and exposure policy

In this paper, we assume that each user i has a privacy policy $P_i(x)$ and a exposure policy $V_i(x)$ for a specific photo x. The privacy policy $P_i(x)$ indicates the set of users who can access photo x and exposure policy $V_i(x)$ indicates the set of users who can access x when user i is involved. After people on co-photo x are recognized with our algorithm as a set $\mathcal{I}$, the set of users who follow both the privacy policy and exposure policy could be calculated by:

$$S = P_i(x) \bigcap_{k \in \mathcal{I}} V_k(x) \tag{1}$$

We assume that our users have defined their privacy policy and exposure policy and these policies are modifiable. The exposure policy is treated as a private data that shall not be revealed, and a secure set intersection protocol [11] is used to find the access policy S in 1. After the access policy S is established, the co-photo x will be shared with users in S.

3.2 FR with social contexts

An FR engine for a large-scale social network may require discriminating millions of individuals. It seems to be a daunting task that could never be accomplished. However, when we decompose it into several personal FR engines, the situation will change for better. Social contexts contains a large amount of useful information which could be utilized as a priori knowledge to help the facial recognition[19]. In [12], Mavridis, Kazmi and Toulis develop a three-realm model to study facial recognition problems on OSN photos. The three realms include a social realm, in which identities are entities, and friendship a relation; a visual sensory realm, of which faces are entities and occurrence in images a relation; and a physical realm, in which bodies belong, with physical proximity being a relation. It is shown that the relationship in the social realm and physical realm are highly correlated with the relationship in the visual sensory realm. In this manner, we can use the social context to construct a priori distribution $\mathcal{P}_i$ over the identities on the co-photos for user i. With this priori distribution, while trying to recognize people on the co-photos, the FR engine could focus on a small portion of "close" friends (friends who are geographically close and interacting frequently with user i).

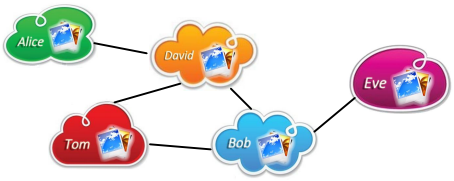


Fig. 1: A friendship graph in visual sensory realm

Fig.1 shows a relational graph in the visual sensory realm. We assume that for user i, we can define a threshold on the priori distribution $\mathcal{P}_i$ to get a small group of identities consisting of i and his one-hop neighbors (e.g., close 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 example, in Fig.1, if Bob has a co-photo, we assume that users appear in the photo are among the set of {Divid, Eve, Tom, Bob}.

3.3 FR system

We assume that $user_i$ has a photo set of size N_i of himself/herself as his/her 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 [23]. For each face, a vector of size p is extracted as the feature vector. Then, for user i, his/her private training set could be written as x_i of size $N_i \times p$. In the rest of this paper, we use one record and one photo interchangeably to refer one row in x_i.

With the private training set, each user will have a personal FR engine to identify his/her one-hop neighbors. The personal FR can be constructed as a multi-class classification system, where each class is corresponding to one user (himself/herself or one friend). In the rest of this paper, we use one class interchangeably with the appearance of one user. In the realm of machine learning, usually a multi-class classification system is constructed by combining several binary classifiers together with the one of the following strategies[7]:

- **One-against-all method** uses winner-take-all strategy. It constructs n binary classifiers for each of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

n classes. The goal of each binary classifier is to distinguish one class from the rest with a decision function. Hence, the ith decision function f_i is trained by taking records from user i as positive samples and the records from all the other users as the negative samples. When a testing record x comes, if f_i concludes that it belongs to class i, x is labeled as class i.

- **One-against-one method** uses max-voting-win strategy. It constructs $n(n-1)/2$ binary classifiers, in which each classifier is aimed to distinguish two classes. The idea is that if we can distinguish any two classes, then we can identify any of them. Hence, classifier u_{ij} is constructed by taking records from i as positive samples and records from j as negative ones. Later on when we are trying to identify a test record x, if u_{ij} concludes that x is in class i, then the vote of class i is added by one. After testing all the $n(n-1)/2$ classifiers, x is assigned to the class with the largest voting value.

However, no matter which method we use, it requires a centralized node to access all the training samples from each class, which is conflicting with our promise that the private training samples will not be disclosed during the whole process. In the rest of this paper we will focus on how to build the personal FR engines without disclosing the private photo sets. Notice that the identification criterion could be asymmetric between different personal FR engines, which means that the way how David finds out Bob and how Bob finds out David are not the same as shown in Fig. 1. The reason is that, for Bob, his personal FR engine only knows how to find out David from the candidate set ("suspects" for short) of {Bob, David, Eve, Tom}, while for David, his personal FR only knows how to find out Bob from the suspects of {Alice, Bob, David, Tom}. In other words, with different friend sets (friendship graph) at each node, the personal FR engines are trained with different negative training samples.

4 System Overview

In this section, we present the detailed description of our system. Generally speaking, the consensus result could be achieve by iteratively refining the local training result: firstly, each user performs local supervised learning only with its own training set, then the local results are exchanged among collaborators to form a global knowledge. In the next round, the global knowledge is used to regularize the local training until convergence. In this section, firstly, 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 users. Finally, we discuss the scalability of our design at the large scale of OSNs.

4.1 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 and we use linear kernel for the ease of presentation. 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 class labels of samples in X_i on its diagonal elements. Let X_1 denote the positive sample set, X_2 the negative sample set and a diagonal matrix Π is constructed as a $(p+1) \times (p+1)$ diagonal matrix with $\Pi(i,i) = 1$ for $i = 1, 2, \ldots, p$ and $\Pi(p+1, p+1) = 0$. Then, the decision function $f(\cdot)$ can be obtained by solving the following problem:

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

In problem (2), by minimizing $\frac{1}{2} u^T \Pi u$, we find u that maximizes the margin between the positive and negative training set. The constraints are used to ensure that the decision function satisfies the training set. 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 assigned to the objective function, where C is the user-chosen penalty parameter and vice versa for the negative samples. Notice that the constraints are private training data which are not available for a centralized SVM solver. Our approach is to split (2) into two subproblems with their own constraints and an additional constraint $u_1 = u_2$ as:

$$\begin{aligned} \min_{u_1,\xi_1 \geq 0} \quad & \frac{1}{4} u_1^T \Pi u_1 + C\|\xi_1\| \\ \text{s.t.} \quad & Y_1 X_1 u_1 \geq 1 - \xi_1, \\ & u_1 = u_2, \end{aligned} \tag{3a}$$

$$\begin{aligned} \min_{u_2,\xi_2 \geq 0} \quad & \frac{1}{4} u_2^T \Pi u_2 + C\|\xi_2\| \\ \text{s.t.} \quad & Y_2 X_2 u_2 \geq 1 - \xi_2, \\ & u_1 = u_2. \end{aligned} \tag{3b}$$

We can easily show that problem (3) is an identical transformation of problem (2) by substituting $u = u_1 = u_2$ and putting together the constraints[8]. Problem (3a) and (3b) could be assigned to $user_1$ and $user_2$ accordingly and be solved by alternatively optimize u_1 and u_2. u_1^t and u_2^t might be very different at the first few iterations, however, they will slowly reach the consensus as t grows.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

To solve this problem, firstly, we need to find the augmented Lagrange function with the Language multipliers of $\{\lambda_i\}$ and $\{\alpha_i\}$ as:

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

In Eq. (4), 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[8].

$\mathcal{L}$ could then be minimized in a cyclic fashion: at each iteration, $\mathcal{L}$ is minimized with respect to one variable while keeping all other variables fixed. According to Alternating Direction Method of Multipliers (ADMM)[3], update of the variables at each iteration $t+1$ could be summarized as follows,

$$\begin{aligned} u_i^{t+1} &= \operatorname*{argmin}_{u_i} \mathcal{L}(u_i, \{u_{j\neq i}^t\}, \{\lambda_i^t\}, \{\alpha_i^t\}); \\ \alpha_i^{t+1} &= \alpha_i^t + \rho(u_i^{t+1} - u_j^{t+1}). \end{aligned} \tag{5}$$

In (5), u_i is calculated through the Wolfe dual problem. User i is could compute u_i^{t+1} locally, because it is only related to X_i, Y_i, λ_i^t and u_j^t but have nothing to do with X_j and Y_j. This data isolation property is the essence of our secure collaborative learning model and the detailed security analysis will be presented in Section 5). With KKT conditions and Wolfe dual, detailed iterative updates are listed in Eq. (6).

$$\begin{aligned} \lambda_i^{t+1} = \ & \operatorname{argmax}\{-\lambda_i^T Y_i X_i (\Pi + 4\rho I)^{-1} X_i^T Y_i \lambda_i \\ & + [1 + 2Y_i X_i (\Pi + 4\rho I)^{-1}(\alpha_i^t - \alpha_j^t - 2\rho u_j^t)]^T \lambda_i\} \\ u_i^{t+1} = \ & 2(\Pi + 4\rho I)^{-1}[X_i^T Y_i \lambda_i^{t+1} - (\alpha_i^t - \alpha_j^t) + 2\rho u_j^t] \\ \alpha_i^{t+1} = \ & \alpha_i^t + \rho(u_i^{t+1} - u_j^{t+1}). \end{aligned} \tag{6}$$

Generally, the proposed distributed training scheme of a toy system could be summarized in Algorithm 1. In this Algorithm, $u_{ij} = F(X_i, X_j)$ is the computation of classifier u_{ij} with X_i as positive training samples and X_j as negative training samples. $\text{qd}(A, B)$ is a standard quadratic programming solver that gives the optimal solution of $\max\{-\frac{1}{2}x^T A x + B^T x\}$, 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 fewer iterations while a larger discrepancy between u_i and u_j. Theorem 4.1 asserts the convergence of Algorithm 1.

Theorem 4.1: By following Algorithm 1, the resulting u_{ij} will converge to the feasible solution u of problem (2) after a certain number of iterations.

Algorithm 1: Iterative Method to Compute u_{ij}

Input: Positive samples X_i, Negative samples X_j
Output: The classifier $u_{ij}(\cdot)$
Initial $\lambda, u_i^0, u_j^0, \alpha_i^0, \alpha_j^0$ 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};

Proof: For the toy system, problem (2) could be written in a general form as follows:

$$\begin{aligned} \min_{v,u} \ & F_1(u_1) + F_2(u_2) \\ \text{s.t.} \ & Au_1 = u_2, \\ & u_1 \in \mathcal{S}_1, \ u_2 \in \mathcal{S}_2. \end{aligned} \tag{7}$$

In problem (7), $F_1(\cdot)$ is the local problem for $user_1$ and $F_2(\cdot)$ is the local problem for $user_2$. A is an identity matrix to ensure that $u_1 = u_2$. It is proved in [8] and [3] that the convergence of problem (7) is guaranteed as long as one of the following two conditions is true: $\mathcal{S}_1$ is bounded; or $A^T A$ is nonsingular. In our scheme, A is an identity matrix, hence, u_1 and u_2 will converge to the same optimal value. □

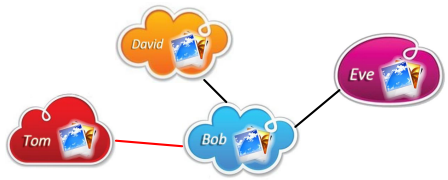
4.2 OSNs with social contexts

In the previous subsection, we show how to build a binary classifier in a toy system with two users. When considering the practical scenario, each user may have more than one friend, and thus multi-class classifiers are required. 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. In this section, we analyze their performance and present mechanisms with the proper strategy.

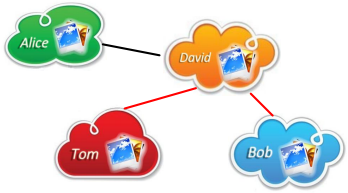
4.2.1 Two strategies and classifier reuse

First, let us introduce some notations: we denote user i as the *initiator* when X_i is used as the positive training samples and user j as the *cooperator* when X_j is used as negative samples. We denote a node i in friendship graph and its one-hop neighbors as $\mathcal{B}_i$: the neighborhood of i. A personal FR engine for user i should be trained to distinguish users in $\mathcal{B}_i$. We use a node i on the friendship graph interchangeably with user i.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing



(a) In the neighborhood of David



(b) In the neighborhood of Bob

Fig. 2: A one-hop neighbor could also be a two-hop neighbor

For the strategy of one-against-all, each user j in $\mathcal{B}_i$ are associated with a binary classifier $f_j(\cdot)$ by making j *initiator* and $\{k \in \mathcal{B}_i, k \neq j\}$ *cooperators*. Denote D_i the degree of user i, there will be D_i+1 classifiers and each classifier involves D_i+1 users. The cost to build one classifier is hence $\mathrm{O}\big(n^\epsilon \mathcal{T}_a \bar{D}\big)$, where $\mathrm{O}\big(n^\epsilon\big)$ is the cost of local SVM training with n training items, $\mathcal{T}_a$ is number of iterations to converge and $\bar{D}$ is the average degree for a node in friendship graph. Hence, total cost in one neighborhood is $\mathrm{O}\big(n^\epsilon \mathcal{T}_a \bar{D}^2\big)$.

For the strategy of one-against-one, $\frac{1}{2}\bar{D}(\bar{D}+1)$ toy systems need to be trained. The cost for each toy system with 2 users is $\mathrm{O}\big(n^\epsilon \mathcal{T}_o\big)$, where $\mathrm{O}\big(n^\epsilon\big)$ is the cost of local SVM training with n training items, $\mathcal{T}_o$ is number of iterations to converge. Hence, total cost in one neighborhood is $\mathrm{O}\big(n^\epsilon \mathcal{T}_o \bar{D}^2\big)$.

Comparing these two strategies, we can see that the only difference is the term of $\mathcal{T}_a$ and $\mathcal{T}_o$, average number of iterations needed to converge for systems with $\bar{D}$ users and 2 users, respectively. Intuitively, $\mathcal{T}_o$ should be much smaller than $\mathcal{T}_a$, because less data is considered. Another factor makes one-against-one strategy appearing is that we could reuse classifiers among mutual friends. For example, in Fig. 2, Tom, Bob and David are mutual friends. When working in David's neighborhood, we need to build classifier of {Tom, Bob} and later on, when working in Bob's neighborhood, we need to build another classifier of {Tom, Bob}. We know that these two classifiers are identical and hence could be reused. The factor of classifier reuse is highly depend on number of complete subgraphs, which seems to be very common over OSNs. According to the data research team from Facebook, the degrees of separation on Facebook is 3.74, meaning that the average distance between any two people is only 3.74 hops on friendship graph. This means friendship graph contains large numbers of complete subgraphs. In comparison, classifiers cannot be reused in the one-against-all strategy because they are trained with different *cooperators*. The procedure to establish classifiers considering classifier reuse is summarized in Algorithm 2.

Algorithm 2: Classifier Computation Algorithm

```
Initial as C_i = ∅, ∀i ∈ N ;
for i ∈ N do
    for j ∈ B_i do
        if u_ij ∉ C_i then
            u_ij = F(X_i, X_j);
            u_ji = −u_ij;
            C_i = {u_ij, C_i}; C_j = {u_ji, C_j};
        end
    end
end
for i ∈ N do
    for k, j ∈ B_i || k ≠ j do
        if u_kj ∉ C_k then
            u_kj = F(X_k, X_j);
        else
            Request u_jk from user j;
        end
        C_i = {u_jk, C_i};
    end
end
```

According to Algorithm 2, there are two steps to build classifiers for each neighborhood: firstly find classifiers of {self, friend} for each node, then find classifiers of {friend, friend}. Notice that the second step is tricky, because the friend list of the neighborhood owner could be revealed to all his/her friends. On the other hand, friends may not know how to communicate with each other. For this consideration, when building classifiers of {friend, friend}, all the local training results are send to the neighborhood owner, who will coordinate the collaborative training processes by forwarding local training results to right collaborators. In this manner, friends need not to know who they are working with and how to talk with them.

4.2.2 Stranger detection

When Algorithm 2 is done, user i is able to differentiate all his friend with classifiers in $\mathcal{C}_i$. The only thing remains to assemble binary classifiers to be a multi-class classifier. In this paper, we construct a decision tree by arranging binary classifiers similarly to the DAGSVM[16]. In the original DAGSVM, the tree nodes contains binary classifiers. Decisions of left or right is made based on output of the tree nodes and class labels are stored at leaf nodes. But a limitation of DAGSVM is that it is based on a strong assumption: *users on a co-photo are friends*, in other words, DAGSVM will always classify x to be one of the friends. In reality, this is not the case, we should be prepared of strangers. For example, Bob has a co-photo with him and Alice at a popular attraction spot. It is very likely that random people could be captured in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

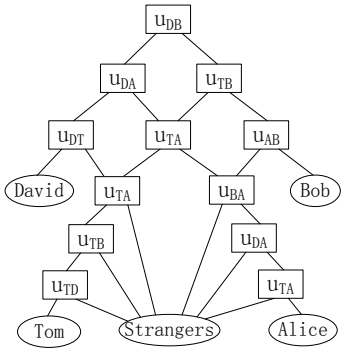


Fig. 3: Improved decision tree

photo. False positive errors could be generated if we try to classify random people as if we know them.

However, detection of strangers (or outliers) is a well-known difficult problem, because there is no such a class of *stranger* has been involved in training. Intuitively, this *stranger* class includes everyone other than these in a certain neighborhood. It requires tons of training samples to construct such a all-embracing class and we simply cannot afford to it. However, we observed that a photo of a *stranger* could make the binary classifiers to output *contradictory decisions*. For example, in Fig.3, a test sample x of Alice cannot make $f_{AD}(x) > 0$ and $f_{TA}(x) > 0$ at the same, otherwise, x belongs to Alice and doesn't belongs to Alice at the same time. Basically, we propose a stranger rejection mechanism based on the following two assumptions: (1) If a certain class participates in the training process, its probing samples never generate *contradictory decisions*. (2) If a certain class does not participate in the training process, its probing samples will make the classifiers to output unpredictable decisions and may result with *contradictory decisions*.

Fig.3 illustrate how DAGSVM is extended to capture *contradictory decisions* by adding more tree nodes. In this extended decision tree, if a probing sample passes all the classifiers of one class, it is assigned to this class, otherwise, it is classified to be a stranger. Theorem 4.2 states correctness of this design.

Theorem 4.2: Assuming a probing sample x is traveling through the DAG decision tree and gets the destination class i, then the only possible class for x to pass all its tests is i.

Proof: The DAG decision tree is based on the exclusive method. Each tree node test will rule out a wrong class. Then, at the leaf node, only one possible class remains. If a sample x exists such that the DAG decision is i while it could pass all the tests of class j, we will get the conflicting result that $f_{ij}(x) > 0$ and $f_{ij}(x) < 0$. Hence, the only possible class of x is the DAG decision class i. □

Notice that the proposed stranger detection scheme brings trivial extra storage cost and computation cost to travel through the tree, which are still $\mathrm{O}(\bar{D}^2)$ and $\mathrm{O}(\bar{D})$, respectively.

4.3 Scalability

To make our system design scalable, we need to consider the following two cases: (1) The private photo set X_i and the corresponding labels Y_i may change over time as X_i^t and Y_i^t. This happens when the appearance of user i has changed, or the photos in the training set are modified (adding new images or deleting existing images). (2) The friendship graph may change over time. For example, when a user moves to another city for work or study, new friends should be added to friendship graph.

For the first case, when X_i is changing, then all the classifiers related to X_i in Algorithm 2 should change. We can modify the iterations in (6) as

$$\begin{aligned}
\lambda_i^{t+1} &= \text{argmax } \{-\lambda_i^T Y_i^{t+1} X_i^{t+1} (\Pi + 4\rho I)^{-1} X_i^{t+1} Y_i^{t+1} \lambda_i \\
&+ [1 + 2Y_i^{t+1} X_i^{t+1} (\Pi + 4\rho I)^{-1} (\alpha_i^t - \alpha_j^t - 2\rho u_j^t)]^T \lambda_i\} \\
u_i^{t+1} &= 2(\Pi + 4\rho I)^{-1} [Y_i^{t+1} X_i^{t+1} \lambda_i^{t+1} - (\alpha_i^t - \alpha_j^t) + 2\rho u_j^t] \\
\alpha_i^{t+1} &= \alpha_i^t + \rho(u_i^{t+1} - u_j^{t+1}).
\end{aligned} \tag{8}$$

In Eq.(8), the private training set X_i now is a variable over time. At each iteration t, local training results are calculated with the current training set X_i^t. Intuitively, the training set X_i varies in a much slower rate than the iterative updates of parameters. In other words, we assume that X_i remain invariant across a sufficient number of iterations, during which the resulting u_{ij} will closely track the optimal classifier between the training sets.

If the social circle of a user is changed, his/her personal FR engine should also be modified. If this modification is made by adding a new friend, new classifiers should be computed or reused by following Algorithm. 2. After that, the existing decision tree could be extended by adding tree nodes with these new classifiers. If the modification is generated by removing the friendship, one just need to remove all the corresponding classifiers and reassemble his/her decision tree.

5 Performance Analysis

In this section, we present the performance analysis of our scheme. In the first subsection, we analyze the computational complexity of our FR system and comparisons with other two possible approaches. In the second subsection, the detailed privacy analysis is presented.

5.1 Benefits of our design

In this subsection, we describe the expected computational complexity of three approaches: centralized solution, one-against-all strategy and our approach. The notations involved are: N is the number nodes and $\bar{D}$ is the average degree of friendship graph, n and p are parameters of private training data X, denote number of training records and length of each training record, respectively.

- **Centralized approach** has a centralized FR engine in charge of recognizing all users over a large OSN.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

To protect the training photos, a privacy-preserving SVM training method [25] is used. In this approach, Yu et al. use secure dot product protocols [9] to evaluate kernel matrix of SVM. The computational cost for the secure dot product protocol based on homomorphic encryption is $\mathrm{O}(p \log m)$, where m is the value of the exponent. SVM kernel matrix is composed of $(Nn)^2$ dot products, hence the total cost is $\mathrm{O}(N^{\epsilon+1}n^{\epsilon}) + \mathrm{O}(N^3 n^2 p \log m)$, where ϵ is a factor between 2 and 3.

- **One-against-all approach** decompose the friendship graph and use our proposed consensus-based training method to perform collaborative training. As we discussed in the previous section, at each iteration, local SVM problem only deals with a training set of size $n \times p$. Hence, computational cost is $\mathrm{O}(\mathcal{T}_a(n^{\epsilon} + n^2 p)) \approx \mathrm{O}(n^{\mathcal{T}_a \epsilon})$. There are $\bar{D}^2$ local training problems to find $\bar{D}$ classifiers for one neighborhood. Hence the total cost for N neighborhoods is $\mathrm{O}(N\bar{D}^2\mathcal{T}_a n^{\epsilon})$.
- **One-against-one:** The analysis of this approach is similar to the one-against-all approach, except that the average rounds in one training process should be much less, due to the fact that there are only two participants instead of $\bar{D}+1$ ones. If we consider the complete subgraph in the friendship graph, the expected cost should be less than $\mathrm{O}(N\bar{D}^2\mathcal{T}_o n^{\epsilon})$.

A theoretical comparison of the three approaches are listed in Table. 1. We can see that the distributed solutions with context information can greatly reduce the computation. Meanwhile, among the two distributed approaches, the proposed approach should be much more efficient than using the one-against-all approach. In Section 6, we will further demonstrate one-against-one strategy is much more efficient than one-against-all strategy with numerical results.

	complexity	Privacy-preserving	Stranger detection
Centralized	$\mathrm{O}(N^{\epsilon+1}n^{\epsilon})$	✓	×
OVA	$\mathrm{O}(N\bar{D}^2\mathcal{T}_a n^{\epsilon})$	✓	×
Our approach	$\mathrm{O}(N\bar{D}^2\mathcal{T}_o n^{\epsilon})$	✓	✓

TABLE 1: Theoretical comparison of the three approaches

5.2 Security analysis

In this paper, private information of a user is considered as his/hers privacy and exposure policies; friend list and the private training data set X_a. In the rest of this subsection, we show how these private information are protected from a semi-honest adversary.

Privacy and exposure policies: In 1, access policy of x is determined by the intersection of owner's privacy policy and co-owners' exposure policy. In [10], Kissner and Song proposed privacy-preserving set operations including set intersection by employing the mathematical properties of polynomials. We can directly adopt their scheme to find the access policy S.

Friend list: Basically, in our proposed one-against-one strategy a user needs to establish classifiers between {self, friend} and {friend, friend} also known as the two loops in Algorithm. 2. During the first loop, there is no privacy concerns of Alice's friend list because friendship graph is undirected. However, in the second loop, Alice need to coordinate all her friends to build classifiers between them. According to our protocol, her friends only communicate with her and they have no idea of what they are computing for.

Friend list could also be revealed during the classifier reuse stage. For example, suppose Alice want to find u_{bt} between Bob and Tom, which has already been computed by Bob. Alice will first query user k to see if u_{kj} has already been computed. If this query is made in plaintext, Bob immediately knows Alice and Bob are friends. To address this problem, Alice will first make a list for desired classifiers use private set operations in [10] to query against her neighbors' classifiers lists one by one. Classifiers in the intersection part will be reused. Notice that even with this protection, mutual friends between Alice and Bob are still revealed to Bob, this is the trade-off we made for classifiers reuse. Actually, OSNs like Facebook shows mutual friends anyway and there is no such privacy setting as "hide mutual friends".

Private training sets: We assume that Alice and Bob in a toy system are semi-honest. They will follow the protocol but are so curious that they store all the exchanged data and try to trace back others' private training sets. 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 is similar. To show the private training sets are secure, we only need to show that during the $\mathcal{T}_o$ rounds of parameter exchanges, an adversary cannot reverse engineer X of the other user. After $\mathcal{T}_o$ rounds of parameter exchange, information available to Alice is $\{u_b^t, \alpha_b^t\}$, for $t = 1...\mathcal{T}_o$. Her goal is to find an $N_b \times (p+1)$ matrix X_b with $N_b \times p$ unknowns. Alice is familiar with the training mechanism and she knows that the parameters at hand have the relationship as follows:

$$A = 2X_b c^{-1} X_b^T, \tag{9}$$

$$X_b^T \lambda = c u_b^t + d, \tag{10}$$

$$B = 1 + X_b c^{-1} d, \tag{11}$$

$$\lambda = \arg \min_{0 \le \lambda \le \frac{C}{2}} \quad \frac{1}{2}\lambda^T A \lambda + B^T \lambda \tag{12}$$

where $c = 2(\Pi + 4\rho I)^{-1}$, $d = \alpha_b^t - \alpha_a^t - 2\rho u_a^t$ could be computed accordingly for each iteration. Notice that the value of λ comes from the quadratic optimization problem (12), in which A is a fixed matrix determined by X_b, B is changing by iterations. We need to show that, with multiple $\{B, u_b\}$ tuples, Alice cannot get any information of X_b. To solve the quadratic optimization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

problem (12), we need to first find its Lagrange function:

$$\mathcal{L} = \frac{1}{2}\lambda^T A\lambda + B^T\lambda + \tau^T(\lambda - \frac{C}{2}) - \nu^T\lambda, \quad (13)$$

where τ and ν are Lagrange multipliers. The solution of problem (12) could be obtained through the KKT conditions:

$$-A\lambda - \tau + \nu = B^T \quad (14)$$

$$\tau^T(\lambda - \frac{C}{2}) = 0 \quad (15)$$

$$\nu^T\lambda = 0 \quad (16)$$

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

With (9) and (10), (14) can be written as

$$-X_b(c \cdot u_b^t + d) - \tau + \nu = B. \quad (17)$$

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 (12) 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 it is hard to make a reasonable guess for Alice. In this way, at each iteration, by revealing N_b equations, $2N_b$ unknowns are introduced. Alice could never have enough equations to find out X_b.

From another point of view, the information available to Alice is that support vectors of Bob are sitting on a p dimension hyperplane (u_b). One support vector could be found by intersecting p such hyperplanes. However, Bob will never tell Alice which hyperplane contain which support vectors, hence, Alice could not form the proper linear equations to solve a support vector. For these non-support vector training samples, the only information for Alice is that those samples are laying on the opposite side of the hyperplane, Alice have no clue of where they are.

6 Evaluation

Our system is evaluated with two criteria: network-wide performance and facial recognition performance. The former is used to capture the real-world performance of our design on large-scale OSNs in terms of computation cost, while the latter is an important factor for the user experience. In this section, we will describe our Android implementation first and then the experiments to evaluate these two criteria.

6.1 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.4 shows the graphical user interface (GUI). A log in/out button could be 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.

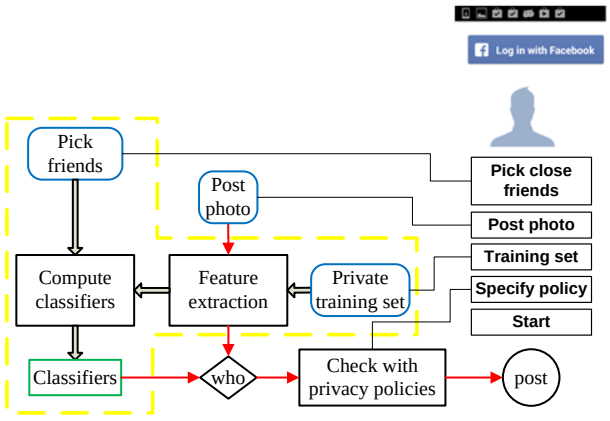


Fig. 4: System structure of our application

Running in the setup mode, the program is working towards the establishment of the decision tree. For this purpose, the private training set X_i and 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 specify the set of "close friends" among their Facebook friends with the button "Pick friends" as their neighborhood. According to the Facebook statistics, on average a user has 130 friends, we assume only a small portion of them are "close friends". In our application, each user picks up to 30 "close 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.4.

During the training process, a socket is established exchange local training results. After the classifiers are obtained, decision tree is constructed and the program switches from the setup mode to the sleeping mode. Facebook allows us to create a list of friends such as "close friends" or "Acquaintances". We can share a photo only to friends on list. According to the proposed scheme, this friend list should be intersection of owner's privacy policy and co-owners' exposure policies. However, in Facebook API, friend lists are read-only items, they cannot be created or updated through the current API. That means we cannot customize a friend list to share a co-photo. Currently, when the button "Post Photo" is pressed, co-owners of x are identified, then notifications along with x are send to the co-owners to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

request permissions. If they all agree to post x, x will be shared on the owner's page like a normal photo. In this sense, users could specify their privacy policy but their exposure policies are either *everybody on earth* or *nobody* depending on their attitude toward x. 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 decision tree will be updated.

6.2 Network-wide performance

At this stage, a large number of users are absent for us to carry out the network-wide evaluation. We simulate a real-life social network with the small-world network[24]. The simulations are conducted on a desktop with Intel i3 550 3.4 GHz and 4.0 GB memory. We use the database of "Face Recognition Data, University of Essex, UK" to assign training set for each simulated users. The database contains photos for 395 individuals and 20 images per individual with varying poses and facial expressions. Users are assigned with photos from the same individual randomly.

In a small world network, there are three input parameters: the total number of vertex N, the average node degree $\bar{D}$ and rewire probability p. In the rest of this section, we use $\bar{D}$ and the number of neighbors interchangeably to denote the average number of users in one's neighborhood. To construct a small-world network, first we arrange the vertices and connect them in a ring. Then we connect every vertex with its $\bar{D}$ nearest neighbors. Finally, for each vertex, with probability p, its existing edge is rewired with another randomly selected vertex. It is shown in [14] that the rewire probability is highly related to the geodesic distance (the average shortest distance between any two vertices). We want to show that in a small-world network, there exist a lot of complete subgraphs, which greatly reduces the setup time by reusing the existing classifiers. Due to resource limitations, we simulate on a network with 3000 vertices. The the computation cost is measured by total computation time.

Fig.5 and Fig.7 plot our simulation results in a network of 3000 nodes with a fixed rewire probability of 0.3 and a varying $\bar{D}$ from 6 to 18. Specifically, as in Fig.5, the one-against-all (OVA) approach and our proposed one-against-one (OVO) approach are compared in terms of total computation cost. We can see that the computation cost of the proposed OVO approach is much lower and the efficiency gain is increasing with number of neighbors. In the previous section, we argued that this phenomenon is caused by two reasons: first, the average number of iterations to converge in our OVO approach should be much smaller; second, the classifiers could be reused with the existence of complete subgraphs.

Fig.6 illustrates the results for the computation cost

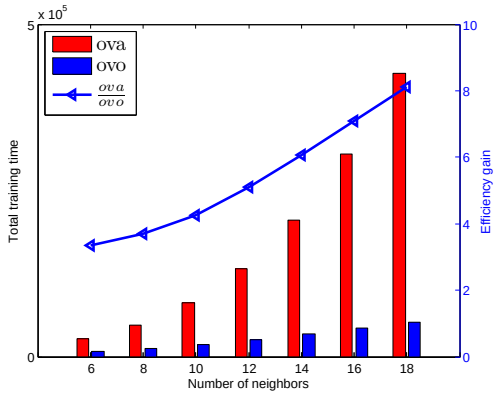


Fig. 5: Total computation cost and the efficiency gain against the number of neighbors

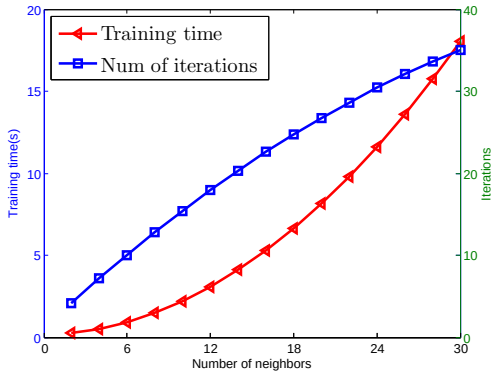


Fig. 6: The average training time and iterations against the number of neighbors

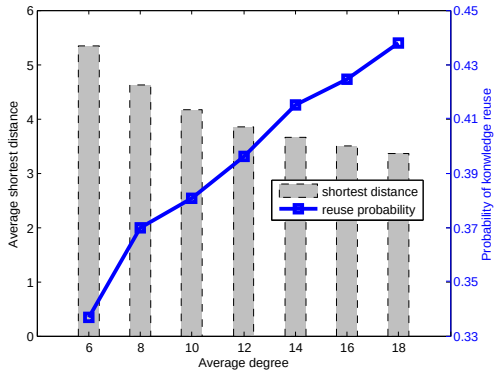


Fig. 7: the average shortest distance and knowledge reuse probability against average degree

and the average number of iterations, which are increasing with the number of participants. In this simulation, each user has 20 training samples and each sample is a vector of 20 features. The stopping criteria is set to be 5%, which means the algorithm will return u_i if its variation is less than 5% between two adjacent iterations. On the one hand, we can see from Fig.6 that for 2 users, it only takes less than 5 iterations to converge, while for 30 users, it takes more than 30 iterations. Moreover,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

for 30 users, each iteration involves 30 users, both the computation and communication cost are much higher than the case where there are only 2 users. As a result, the training time in total for 30 users is 100 times more than that for 2 users.

The probability of classifier reuse is studied in Fig.7, in which we plot the probability of reuse together against the average shortest distance. By reusing a classifier, we mean that when user i and user j attempt to compute a classifier u_{ij}, instead of conducting the iterative algorithm immediately, they first try to look up at the local table. If u_{ij} exists in the table, this classifier could be reused. Fig.7 shows that with a small average shortest distance, the reuse probability is high because a smaller distance between vertices means the vertices are "well connected", in which a complete subgraph is more likely to exist.

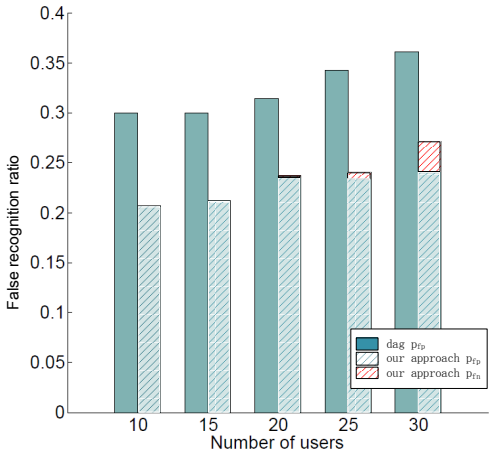


Fig. 9: The false negative and false positive ratios

6.3 Facial recognition performance

In this subsection, we study the recognition ratio against the number of friends and the number of strangers. Standard face detection in [23] is used for face detection and eigenface [22] is used 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. To address this, we assume principle components have already been extract to form a vector space $\mathcal{S}$. User's facial photos are projected into this space as feature vectors. Based on our 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. 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.

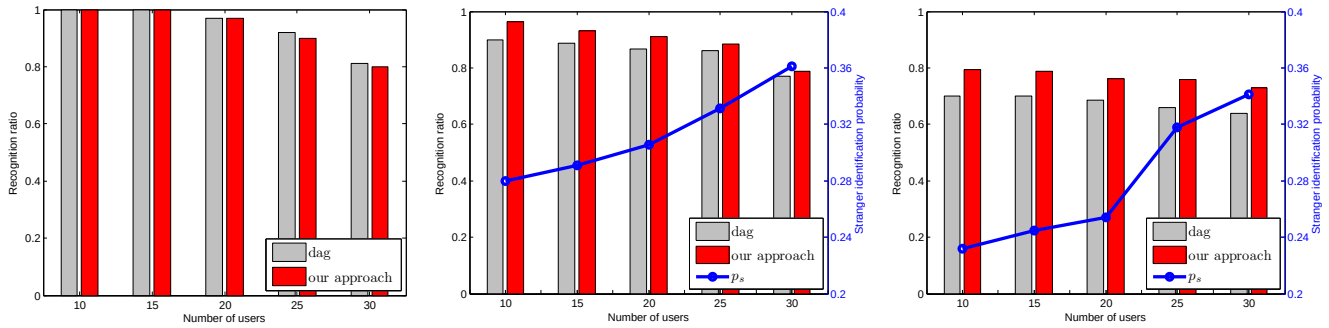
In Fig.8, we show the recognition ratios of our proposed scheme and the scheme with DAG decision tree. As in Fig.8(a), when there are no strangers, both our proposed scheme and the DAG scheme could achieve very high recognition ratio of more than 80% when the number of users is fewer than 30. While in Fig.8(b), among the users, 10% of them are strangers, we can see that the recognition ratio of our scheme has a higher recognition ratio than the DAG scheme by 5%. The reason is that our scheme is able to reject strangers. The solid line on each figure represents recognition ratio of strangers p_s, which is increasing with number of users. Intuitively, if there are more users, there will be more classifiers and the chance that a stranger gets contradictory decisions will be higher. Fig.8(c) shows a similar case where there are 30% strangers. In this case, our scheme outperforms 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%.

Another criterion to measure the performance is the false positive rate. In the previous section we argued that a false positive recognition will reveal the test image to the wrong person. Thus, a low false positive rate is desirable. If there are no strangers, the false positive rate is only determined by the recognition accuracy. If there are strangers, the false positive is also determined by misclassification of the strangers. Fig. 9 illustrates both false positive rate and false negative rate of our scheme and the DAG scheme. We observe that false positive rate of our scheme is 10% lower than original DAG scheme on average. Notice that false negative recognitions could also be introduced by our stranger detection scheme, according to Fig. 9, the more users, the higher chance a user is recognized as a stranger.

7 Conclusion and Discussion

Photo sharing is one of the most popular features 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 proposed to enable individuals potentially in a photo to give the permissions before posting a co-photo. We designed a privacy-preserving FR system to identify individuals in a co-photo. The proposed system is featured with low computation cost and confidentiality of the training set. Theoretical analysis and experiments were conducted to show effectiveness and efficiency of the proposed scheme. We expect that our proposed scheme be very useful in protecting users' privacy in photo/image sharing over online social networks. However, there always exist trade-off between privacy and utility. For example, in our current Android application, the co-photo could only be post with permission of all the co-owners. Latency introduced in this process will greatly impact user experience of OSNs. More over, local FR training will drain battery quickly. Our future work could be how to move the proposed training schemes to personal clouds like Dropbox and/or icloud.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing



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

Fig. 8: Recognition ratio with varying number of users

REFERENCES

[1] I. Altman. Privacy regulation: Culturally universal or culturally specific? *Journal of Social Issues*, 33(3):66–84, 1977.

[2] 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.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011.

[4] 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.

[5] J. Y. Choi, W. De Neve, K. Plataniotis, and Y.-M. Ro. Collaborative face recognition for improved face annotation in personal photo collections shared on online social networks. *Multimedia, IEEE Transactions on*, 13(1):14–28, 2011.

[6] 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.

[7] K.-B. Duan and S. S. Keerthi. Which is the best multiclass svm method? an empirical study. In *Proceedings of the 6th international conference on Multiple Classifier Systems*, MCS'05, pages 278–285, Berlin, Heidelberg, 2005. Springer-Verlag.

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

[9] B. Goethals, S. Laur, H. Lipmaa, and T. Mielik?inen. On private scalar product computation for privacy-preserving data mining. In *In Proceedings of the 7th Annual International Conference in Information Security and Cryptology*, pages 104–120. Springer-Verlag, 2004.

[10] L. Kissner and D. Song. Privacy-preserving set operations. In *IN ADVANCES IN CRYPTOLOGY - CRYPTO 2005, LNCS*, pages 241–257. Springer, 2005.

[11] L. Kissner and D. X. Song. Privacy-preserving set operations. In V. Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 241–257. Springer, 2005.

[12] N. Mavridis, W. Kazmi, and P. Toulis. Friends with faces: How social networks can enhance face recognition and vice versa. In *Computational Social Network Analysis*, Computer Communications and Networks, pages 453–482. Springer London, 2010.

[13] 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.

[14] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[15] L. Palen. Unpacking privacy for a networked world. pages 129–136. Press, 2003.

[16] 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.

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

[18] 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.

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

[20] 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.

[21] K. Thomas, C. Grier, and D. M. Nicol. unfriendly: Multi-party privacy risks in social networks. In M. J. Atallah and N. J. Hopper, editors, *Privacy Enhancing Technologies*, volume 6205 of *Lecture Notes in Computer Science*, pages 236–252. Springer, 2010.

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

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

[24] D. J. Watts and S. H. Strogatz. Collective dynamics of "small-world" networks. *nature*, 393(6684):440–442, 1998.

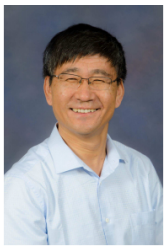
[25] H. Yu, X. Jiang, and J. Vaidya. Privacy-preserving svm using nonlinear kernels on horizontally partitioned data. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 603–610, New York, NY, USA, 2006. ACM.

Kaihe Xu received his B.E. degree from Nanjing university of Posts and Telecommunications, China, in 2010, the M.S. degree from Illinois Institute of Technology, USA, in 2012. He is currently working toward the PhD degree in department of Electrical and Computer Engineering, University of Florida. His research interests include Secure multi-party computation, machine learning and distributed system.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2015.2443795, IEEE Transactions on Dependable and Secure Computing

Yuanxiong Guo received his B.E. degree in electronic information science and technology from Huazhong University of Science and Technology, Wuhan, China, in 2009. He received M.S. and Ph.D. degree in Electrical and Computer Engineering from the University of Florida in 2012 and 2014, respectively. He has been an assistant professor in the School of Electrical and Computer Engineering, Oklahoma State University from August 2014. His research interests include Smart Grids, Power and Energy Systems, Cyber-Physical Systems, and Sustainable Computing and Networking Systems.

Linke Guo received his B.E. degree in electronic information science and technology from Beijing University of Posts and Telecommunications in 2008. He received M.S. and Ph.D. degree in Electrical and Computer Engineering from the University of Florida in 2011 and 2014, respectively. He has been an assistant professor in the Department of Electrical and Computer Engineering, Binghamton University, State University of New York from August 2014. His research interests include network security and privacy, social networks, and applied cryptography. He has served as the Technical Program Committee (TPC) members for several conferences including IEEE INFOCOM, ICC, GLOBECOM, and WCNC. He is a member of the IEEE and ACM.



Yuguang Fang (F'08) received a Ph.D. degree in Systems Engineering from Case Western Reserve University in January 1994 and a Ph.D degree in Electrical Engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at University of Florida in May 2000 as an assistant professor, got an early promotion to an associate professor with tenure in August 2003 and a professor in August 2005. He has published over 350 papers in refereed professional journals and conferences. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He won the Best Paper Award at IEEE ICNP2006. He has served on many editorial boards of technical journals including IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing, Wireless Networks, and IEEE Wireless Communications (including the Editor-in-Chief). He is also serving as the Technical Program Co-Chair for IEEE INFOCOM2014. He is a fellow of the IEEE.

Xiaolin Li is Associate Professor in Department of Electrical and Computer Engineering at University of Florida. His research interests include Parallel and Distributed Systems, CyberCPhysical Systems, and Network Security. His research has been sponsored by National Science Foundation (NSF), Department of Homeland Security (DHS), and other funding agencies. He is an associate editor of several international journals and a program chair or co-chair for over 10 international conferences and workshops. He is on the executive committee of IEEE Technical Committee on Scalable Computing (TCSC) and served as a panelist for NSF. He received a Ph.D. degree in Computer Engineering from Rutgers University, USA. He is the founding director of the Scalable Software Systems Laboratory (http://www.s3lab.ece.ufl.edu). He received the National Science Foundation CAREER Award in 2010. He is a member of IEEE and ACM.