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

**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.

**OBJECTIVES**

A survey was conducted in to study the effectiveness of the existing countermeasure of un-tagging and shows that this

countermeasure is far from satisfactory users are worrying about offending their friends when un-tagging. 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.

**PROBLEM DEFINITION**

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 multiparty computation problem.

Intuitively, we may apply cryptography technique to protect the private photos, but the computational and

communication cost may pose a serious problem for a large OSN.

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.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 This data isolation property is the essence of our secure

collaborative learning model and the detailed security analysis.

**BASIC FUNCTIONALITIES**

**Modules Description:**

**Functions Of Users**

1. View users

2. View Complaints

3. View Reviews

* Registration
* Login
* View profile
* Edit Profile
* Search for friends
* Sent friend request
* View friend request and confirm
* Sent post
* Includes face recognition
* Face Classifier algorithm
* Masking Filling face portions
* Sent notification to those persons who are in the post
* View notification from system and approve/reject post
* Set privacy settings

**SOFTWARE REQUIREMENT SPECIFICATION**

**Hardware Specifications**

The selection of hardware is very important in the existence and proper working of any software. Then selection

hardware, the size and capacity requirements are also important.

* Processor : Intel Pentium Core i3 and above
* Primary Memory : 4GB RAM and above
* Storage : 320 GB hard disk and above
* Display : VGA Color Monitor
* Key Board : Windows compatible
* Mouse : Windows compatible

**Software Specifications**

One of the most difficult task is selecting software for the system, once the system requirements is found out

then we have to determine whether a particular software package fits for those system requirements. The application

requirement:

* Front end : python
* Back end : MySQL
* Operating system : windows 7 or above
* IDE : Pycharm

**ANDROID**

**Hardware Requirements**

A mobile phone with **Android** operating system

Version: Android 2.2 or above

RAM : 1GB

**Software Requirements:**

Platform - WINDOWS/ANDROID/LINUX

Front End - Java, XML (Android Development Tool)

IDE - Android Studio or Eclipse