HideMe: Privacy-Preserving Photo Sharing on Social Networks

Fenghua Li[†‡], Zhe Sun[†‡], Ang Li[§], Ben Niu[†*], Hui Li[¶] and Guohong Cao[‖]
[†]Institute of Information Engineering, Chinese Academy of Sciences, China
[‡]School of Cyber Security, University of Chinese Academy of Sciences, China
[§]Department of Computer Science and Computer Engineering, University of Arkansas, USA
[¶]State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, China
[‖]Department of Computer Science and Engineering, The Pennsylvania State University, PA, USA
[†]{lfh, sunzhe, niuben}@iie.ac.cn, [§]angli@uark.edu, [¶]lihui@mail.xidian.edu.cn, [‖]gcao@cse.psu.edu
[*]Corresponding Author

***Abstract*—Photo sharing on Online Social Networks (OSNs) has become one of the most popular social activities in our daily life. However, some associated friends or bystanders in the photos may not want to be viewed due to privacy concerns. In this paper, we propose the design, implementation and evaluation of HideMe, a framework to preserve the associated users' privacy for online photo sharing. HideMe acts as a plugin to existing photo sharing OSNs, and it enables the following: a) extraction of factors when users upload their photos, b) associated friends in the uploaded photos are able to set their own privacy policies based on scenarios, instead of a photo-by-photo setting, c) any user in other friend's uploaded photos could be hidden away from unwanted viewers based on one time policy generation. We also design a distance-based algorithm to identify and protect the privacy of bystanders. Moreover, HideMe not only protects users' privacy but also reduces the system overhead by a carefully designed face matching algorithm. We have implemented a prototype of HideMe, and evaluation results have demonstrated its effectiveness and efficiency.**

I. Introduction

With the development of Online Social Networks (OSNs) and smartphones, mobile users can take photos anytime and anywhere, and share them in social communities easily. For example, there are over 350 million photos uploaded daily on Facebook[1], and users share over 700 million photos daily on Snapchat[2].

As more and more people enjoy the benefit of photo sharing, privacy has become a major concern. Most existing photo sharing sites (e.g., Instagram, Flicker, Facebook and WeChat, etc.) allow registered users to access others' photos with limited constraint or no constraint. Although some access control mechanisms are employed to control the access of information contained in their own spaces, users have no control over photos residing outside their spaces; i.e., although your face is in a photo taken by someone else, you cannot control how the photo is shared, and this is most likely decided by the photo taker or uploader in most OSNs such as Facebook [7]. Such model may raise serious privacy concerns through a phenomenon called "Friend-of-a-Friend" leakage model, in which the privacy information of a specific user is revealed by his/her friends. A related event is the information leakage of Cambridge Analytica, the data analytics firm that involved in the US president election in 2016, which amassed a trove of Facebook user data for 87 million people without getting their permissions[3]. Similar privacy leakage exists in other scenarios related to co-location [3]. From the privacy perspective, some unwanted viewers could launch attacks based on these vulnerabilities. At the same time, users also care about how others perceive and interact with their photos [1]. Therefore, we have to preserve user privacy in photo sharing.

To preserve privacy in photo sharing, most existing work [20], [29], [13], [27], [6] focused on designing access control-based approaches, and little work considers the specific sharing scenario. These existing work [8], [7], [22] requires users to set privacy policy for each photo and hence is not scalable. Moreover, since setting privacy policy for each photo is tedious and time-consuming, many users may opt not to share photos, thus missing potential social opportunities. Some other researchers focus on preserving the privacy of the bystanders by using some special equipment or tags [11] [16], which may not be practical.

In this paper, we develop a scenario-based photo sharing framework called HideMe, for privacy-aware users. HideMe preserves users' privacy with a scenario-based access control, where the scenario is defined based on various contexts related to temporal, spatial, and sharing behavior factors. It protects the bystanders' privacy with a distance-based algorithm, and speeds up the face matching process with a pre-matching algorithm. The main contributions of this paper are summarized as follows:

- We conduct a case study on photo sharing preferences from 487 participants and characterize their sharing behavior based on four factors: *temporal*, *spatial*, *interpersonal* and *attribute*.
- We propose the design, implementation and evaluation of HideMe, which allows users to build a scenario-based

[1]http://www.businessinsider.com/facebook-350-million-photos-each-day-2013-9

[2]http://www.theverge.com/2014/5/1/5670260/real-talk-the-new-snapchat-makes-texting-fun-again-video-calls

[3]http://www.bbc.com/news/technology-43649018

access control model by combining the above factors, and then decide to blur/show their faces to photo-viewers for each scenario.
- We design a practical algorithm to utilize the focal length information in photography to calculate the photographic distance, which is used to protect the privacy of the bystanders.
- We design an efficient face matching algorithm to be applicable for large scale databases. It relies on pre-matching based on four carefully chosen facial attributes, to filter out a large user database to a small set of candidates. Then, the face recognition is only performed on a small set of candidates to speed up the matching process.

The rest of this paper is organized as follows. In Sec. II, we review the related work. Sec. III presents some preliminaries. Sec. IV and Sec. V describe the proposed HideMe framework and its implementation, respectively. Sec. VI presents performance evaluation results. Finally, Sec. VII concludes the paper.

II. Related Work

As a typical instantiation of Privacy Computing [12], privacy preserving Photo sharing has received considerable attention [19], [26], [4], [14]. Hu *et al.* [7] addressed the privacy conflicts by quantifying privacy risk and sharing loss based on the tradeoff between privacy preservation and data sharing. Subsequently, Ilia *et al.* [8] addressed the privacy conflicts by changing the granularity of access control from photo level to face level. Vishwamitra *et al.* [22] improved Ilia's scheme to control more objects instead of only faces. To improve efficiency, Xu *et al.* [25] employed a distributed consensus-based method, and designed a facial recognition method to identify users in the photos. There are also some work which focus on the privacy of the metadata and facial feature in photo sharing [30], and they are easily integrated into our framework. However, these solutions to privacy conflicts are hard to be widely deployed since they require users to set privacy policy for each photo. In our work, we try to avoid this problem by setting privacy policy at the scenario level instead of photo/face level.

Although there are lots of research on preserving the bystanders' privacy [21], [28], [11], [2], they are mainly at the photo taken phase. Pallas *et al.* [16] addressed this problem from a different perspective. They designed Offlinetags based on worn tags, and their mechanism could respect the users' preferences when the photo is shared. Unfortunately, it is difficult for users to wear such Offlinetags anywhere and anytime. We try to solve this problem by using photographic distance to identify bystanders.

Although there has been a large amount of research on using face recognition to achieve better control of photo sharing [17], [15], most of them cannot be applied for large scale scenarios. To address this problem, Toubiana *et al.* [21] used location information to filter users who frequently appear at the same place. However, this method introduces location privacy risks, especially in location-based services. Xu *et al.* [25] exploited the relationship information to improve the efficiency of face matching. However, it is hard to apply this solution to deal with bystanders. Wang *et al.* [24] proposed a deep feature based scheme to filter a large face database to a small set of candidate face images, but it is challenging to find the optimal size of the candidate set.

TABLE I
Factors on whether to share photo

Factors	China	USA	Total
Temporal Factor			
Weekdays 8:00-18:00	84(24.8%)	18(12.2%)	102(20.9%)
Weekdays 18:00-8:00	142(41.9%)	44(29.7%)	186(38.2%)
Weekends	**210(61.9%)**	**122(82.4%)**	**332(68.2%)**
Holidays	**207(61.1%)**	**116(78.4%)**	**323(66.3%)**
Spatial Factor			
Daily outdoor activity	115(33.9%)	13(8.8%)	128(26.3%)
Restaurant	72(21.2%)	24(16.2%)	96(19.7%)
Private gathering(e.g. party)	**237(69.9%)**	**113(76.4%)**	**350(71.9%)**
Worship	76(22.4%)	79(53.4%)	155(31.8%)
Bar or nightclub	95(28.0%)	84(56.8%)	179(36.8%)
Gym	60(17.7%)	24(16.2%)	84(17.2%)
Public transit	54(15.9%)	22(14.9%)	76(15.6%)
Workplace	159(46.9%)	83(56.1%)	242(49.7%)
Hospital	119(35.1%)	**119(80.4%)**	238(48.9%)
Public gathering(e.g. movie)	58(17.1%)	33(22.3%)	91(18.7%)
Other	15(4.4%)	7(22.3%)	22(4.5%)
Total # of Responses	339(100%)	148(100%)	487(100%)

III. Preliminaries

A. User Study

To better understand the sharing behavior and privacy preference of users on photo sharing, we conducted a survey with two stages. In the first stage, we publicized a basic questionnaire of spatio-temporal factors in University of Arkansas. In the second stage, we improved our survey and added more questions based on the feedbacks in the first stage, and published it in China[4] (N_1 = 339) and United States[5] (N_2 = 148), respectively. We received 487 responses totally for the basic part and 373 responses for the extended part. The results are summarized in Tab. I and Tab. II.

The results show that four factors affect the sharing decisions. They are *temporal*, *spatial*, *interpersonal* and *attribute* factors, which correspond to "when", "where", "who" and "what", part of the "5W1H or Six Ws" [5] in information gathering or problem solving.

Temporal Factor (When). The photo shooting time is very important for users to decide whether the photos are private. For 68.2% of weekends and 66.3% of holidays, some participants explicitly mentioned "Any time of private life is sensitive".

Spatial Factor (Where). Photo shooting location is important. It can be further classified by Point of Interests (PoIs), because different locations lead to different PoIs. As shown in Tab. I, PoI is highly personalized for users, and the responses are distributed in a decentralized fashion. However, more participants (69.9% in China and 76.4% in US) care about the privacy of private gathering (e.g., party). We also notice

[4]https://www.wenjuan.com/s/AvMjQbI
[5]https://goo.gl/forms/GVN6upAtPzQr4R9p2

TABLE II
EXTENSIONAL FACTORS ON WHETHER TO SHARE PHOTO

Factors	China	USA	Total
Interpersonal Factor			
Special name list	133(39.2%)	19(55.9%)	152(40.8%)
Relationships	173(51.0%)	15(44.1%)	188(50.4%)
Circles	**208(61.4%)**	**15(44.1%)**	**223(59.8%)**
Attribute Factor			
# People in a photo ≤ 5	**206(60.8%)**	**21(61.8%)**	**227(60.9%)**
≤ 6	50(14.8%)	3(8.8%)	53(14.3%)
≤ 7	18(5.3%)	1(2.9%)	19(5.1%)
≤ 8	45(13.3%)	4(11.8%)	49(13.2%)
Photographic distance $\geq$ 4m	**159(46.9%)**	**18(52.9%)**	**177(47.5%)**
$\geq$ 5m	91(26.8%)	5(14.7%)	96(25.7%)
$\geq$ 6m	33(9.7%)	1(2.9%)	34(9.1%)
$\geq$ 7m	42(12.4%)	8(23.5%)	50(13.4%)
Total # of Responses	339(100%)	34(100%)	373(100%)

that participants in US care more about the privacy of hospital visits (80.4%).

Interpersonal Factor (Who). Three categories are used to learn the user preference for interpersonal management, which includes a) specified name list, b) relationship (kinship, friendship, friend of a friend, stranger, etc.), and c) friend circles. There is no determining one that meets all the users' requirements, but the friend circle (59.8%) has more weight over others.

Attribute Factor (What). Two photo attributes are widely accepted as factors that affect user sharing decisions. The first one is the number of users in the photo since it can be used to infer the sensitivity in most scenarios. A common example is that people may not worry too much about privacy with a group picture. The other one is the photographic distance which can be used to identify the bystanders. Most participants believe that a person certain distance away from the camera can be considered as a bystander. In our survey, most participants prefer more stringent rules on these two attributes.

We also receive some positive feedbacks that participants prefer to use some additional factors to control photo sharing, such as intimacy of viewers, co-occurring users in the photos, thus, they can be considered into the Interpersonal Factor and Attribute Factor respectively.

B. Motivation and Basic Idea

Our work is motivated by a set of observations in our daily life. **Observation I**: Although different people have different privacy concerns in different scenarios for each specific photo, they prefer less number of privacy settings for photo sharing. **Observation II**: Personalized privacy policy for each shared photo in each scenario is a good solution to preserve user's privacy. However, this is hard to achieve since users do not want to spend a large amount of time on setting up privacy policies for each photo. **Observation III**: There are few practical solutions to preserve bystander's privacy in photo sharing.

In Aditya *et al.*'s [2] survey[6], 227 responses from 32 countries were collected to reach the following conclusions: a) privacy policies should be individualized since even in the same scenario, different people may have different privacy concerns, b) privacy policies should be situational since people would make different choices in different contexts, places, events and social relationships, c) compliance by courtesy is sufficient since most people have the same opinion that they would obey the privacy wishes of the associated friends and bystanders.

[6]http://goo.gl/forms/6tGG0YmFFG

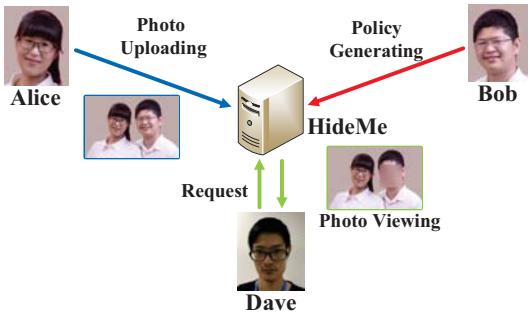


Fig. 1. Use case of HideMe

Based on our observations and conclusions from the survey, the idea behind HideMe is to explore new factors to build specific scenarios, e.g., using the aforementioned "when", "where", "who" and"what" factors. To maximize the number of photos covered by each privacy policy while maintaining individual privacy, HideMe provides flexible privacy control by three main phases. First, HideMe identifies all the faces in each uploaded photo and extracts some effective factors from the metadata information, then it calculates the photographic distance based on these factors to protect bystanders' privacy. Second, HideMe builds a specific scenario for each privacy policy by combining temporal, spatial, interpersonal and attribute factors. Third, when a photo-viewer requests for a specific photo, the carefully designed access control model can hide the privacy-aware users' faces from unwanted photo viewers.

IV. PRIVACY-PRESERVING PHOTO SHARING

A. System Overview

Fig. 1 illustrates how HideMe is used, where users have three roles.

Photo-uploader: the user who uploads the photos. For example, Alice uploads and shares her photos in OSNs.

Policy-generator: the user who generates privacy policies to preserve his/her privacy. For example, Bob is an associated friend appeared in Alice's uploaded photos, he can generate a policy such as "only my friend can access my photos". The photo-uploader Alice can also be a policy-generator, generating her own privacy policy.

Photo-viewer: the user who has permission to see the uploaded photos on the photo sharing-supported OSNs freely. For example, Dave is a photo-viewer, who tries to access a shared photo by Alice. Suppose Dave is Alice's friend, but not Bob's. Bob does not want to show his face in the uploaded photo to Dave, according to his policy. Therefore, HideMe will blur Bob's face before showing it to Dave to avoid the privacy problems in the "Friend-of-a-Friend" leakage model. That's

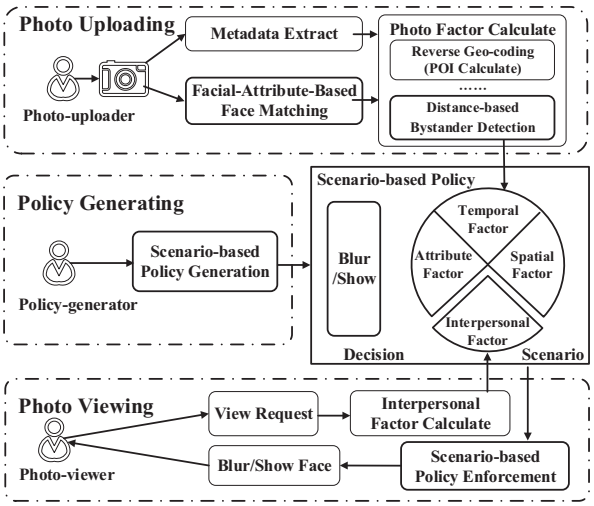


Fig. 2. Data flow of HideMe

also means to meet the privacy wishes of the associated friends and bystanders as discussed in Sec. III-B, our HideMe can enforce any policy-generators' polices without any additional operations of photo-uploaders.

B. System Design

As shown in Fig. 2, HideMe has three phases: photo uploading, policy generating and photo viewing.

1) Photo Uploading: Once the photo-uploader uploads a photo, HideMe extracts the metadata, processes the face information and computes the aforementioned factors in turn.

Firstly, HideMe extracts information from the metadata (Exchangeable Image File, EXIF) including date, time, longitude, latitude, 35mm equivalent focal length and digital zoom ratio, etc. Then, all the faces in the photo are detected, and the relative pixel coordinates and size are obtained and stored. We design a facial-attribute-based face matching module to meet the demands of large scale users. Some information such as date, time, longitude and latitude can be employed as factors directly, and others can be used to calculate factors like PoIs and Photographic Distance. The distance information can be used to detect bystanders.

2) Policy Generating: Each associated friend in the photo has the right to decide whether to blur/show his/her face or not. However, setting policies for each photo may be hard and time wasting. Therefore, HideMe builds a scenario for each policy-generator in order to help them to choose to blur/show their faces, instead of setting policies photo-by-photo. Specifically, when a photo is uploaded, HideMe identifies all faces and obtains the relative user IDs from OSNs, and then associates the corresponding scenarios with this photo. In this way, the policy-generators do not need to set policies for each photo.

3) Photo Viewing: When a photo is requested by a photo-viewer, HideMe processes the policies on all detected faces in the photo. Specifically, HideMe first searches the interpersonal factors between the photo-viewer and each policy-generator, and calculates them when the request comes from a new viewer to the policy-generators. The scenario-based access

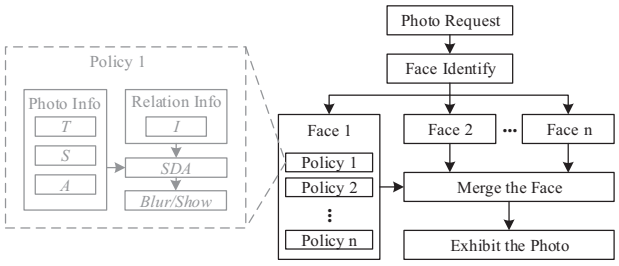


Fig. 3. Information flows for scenario-based access control model

control (describe in the next section) selects the corresponding authorizations of the policies in parallel and enforces it based on user preferences. As shown in Fig. 3, HideMe merges all the faces in one photo and shows it to the photo-viewer.

C. Scenario-based Access Control

In HideMe, permissions can be acquired through scenarios, which are related to the context that a photo-viewer can obtain from a photo. Therefore, scenario (SC) is defined based on the factors and their constraints illustrated in Tab. III. Temporal, spatial, interpersonal, and attribute factors are denoted as T, S, I, and A, respectively.

Scenario Constraints are used to specify the exact condition that a photo is shown or blurred to a particular viewer. It can be built based on any combination of the factors shown in Tab. III. The factor constraints have the general form of $([(V,P)], X_Y, D)$, where (V,P) specifies a request from the viewer to the photo. X can be the factor constraints: T, S, I and A. Y is a specific factor in X. D represents the decision of the photo request, in which we use *show/blur* as an example. The square bracket in $[(V,P)]$ implies that this parameter is optional. When $[(V,P)]$ is selected as a default value, it means the policy will be applied to all requests. Accordingly, the scenario can be termed as $sc = (t, s, i, a)$, where $t \subseteq T, s \subseteq S, i \subseteq I, a \subseteq A$, and $i \neq \emptyset$.

Therefore, HideMe can be constructed as an access control model as follows:

- $(V,P) = \{(viewer_1, photo_1), \cdots, (viewer_n, photo_m)\}$
- $T = \{T_{duration},\ T_{periodcity}\}$
- $S = \{S_{GPS},\ S_{POI}\}$
- $I = \{I_{blacklist},\ I_{relationship},\ I_{circle},\ I_{intimacy}\}$
- $A = \{A_{participants},\ A_{co-occur},\ A_{distance}\}$
- $D = \{show,\ blur\}$
- $SC \subseteq 2^T \times 2^S \times 2^I \times 2^A$
- $VSA \subseteq V \times SC$, a many-to-many mapping (V,P)-to-scenario assignment relation
- $SDA \subseteq SC \times D$, a many-to-many mapping scenario-to-decision assignment relation.

For each scenario, we define the Viewer-Scenario Assignment as *VSA* and Scenario-Decision Assignment as *SDA*. For better understanding, we show an example of scenario in which a user does not want to show his photos taken in Hawaii last summer during holiday nights to his co-workers. When a view request is queried from his co-worker David to *Photo1*, the policy can be model as

TABLE III
CONSTRAINT CATEGORIES OF FACTORS

Constraint Categories / Factors		Expressions	Examples
Temporal Factor	Duration	$([(V,P)], T_{duration}, show/blur)$	$T_{duration} = \langle time_{start}, time_{end}\rangle$,e.g.“$\langle 20180321, 20180520\rangle$”
	Periodicity	$([(V,P)], T_{periodcity}, show/blur)$	$T_{periodcity} = \langle time_{start}, time_{end}, periodcity\rangle$, e.g.“$\langle 20:00:00, 04:00:00, everyday\rangle$”
Spatial Factor	Precise Location	$([(V,P)], S_{GPS}, show/blur)$	$S_{GPS} = \langle center_{longitude}, center_{latitude}, radius_{max}\rangle$, e.g.“$\langle 19°46'06.90''N, 155°33'42.74''W, 20km\rangle$”
	PoI	$([(V,P)], S_{POI}, show/blur)$	$S_{POI}] = \langle POI_1, ..., POI_n\rangle$,e.g.“$\langle hospital, bar\rangle$”
Interpersonal Factor	Per User	$([(V,P)], I_{userlist}, show/blur)$	$I_{userlist} = \langle Name_1, ..., Name_n\rangle$,e.g.“$\langle Bob, David\rangle$”
	Group Users	$([(V,P)], I_{relation}, show/blur)$	$I_{relation} = \langle Relation_1, ..., Relation_n\rangle$,e.g.“$\langle Friend\rangle$”
		$([(V,P)], I_{circle}, show/blur)$	$I_{circle} = \langle Circle_1, ..., Circle_n\rangle$,e.g.“$\langle Workmate\rangle$”
	Intimacy	$([(V,P)], I_{intimacy}, show/blur)$	$I_{intimacy} = Intimacy_{min}$,e.g.“$< 70\%$”
Attribute Factor	# of Faces	$([(V,P)], A_{faces}, show/blur)$	$A_{faces} = Faces_{min}$,e.g.“$< 3\ faces$”
	Co-occurring Users	$([(V,P)], A_{co-occur}, show/blur)$	$A_{co-occur} == \langle Name_1, ..., Name_n\rangle$,e.g.“$\langle Alice\rangle$”
	Photographic Distance	$([(V,P)], A_{distance}, show/blur)$	$A_{distance} = Distance_{max}$,e.g.“$> 3\ meters$”

$((David, Photo1), \{\{\langle 2017/07/23, 2017/08/17\rangle, \langle 20:00:00, 04:00:00, everyday\rangle\}, \langle 19°46'06.90''N, 155°33'42.74''W, 20km\rangle, co-worker, \emptyset\}, blur)$.

Similarly, the user can build another scenario to avoid sharing of the bystanders. Suppose he sets the photographic distance to 4m, which means that the bystanders outside 4m from the camera lens will not be shown to the viewers. It can be modeled as $((V_{default}, P_{default}), \{\emptyset, \emptyset, Everyone, 4m\}, blur)$.

D. Distance-based Bystander Detection

We use the focal length to calculate the photographic distance, which is used to detect the bystanders.

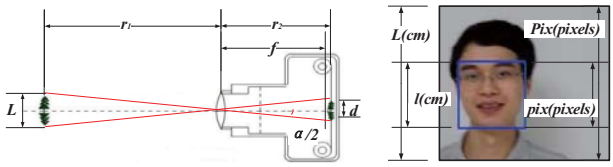
Photographic Distance. In photography, Angle of View (AoV) describes the angular extent of a given scene that is taken by a camera, and it can be used to estimate suitable photographic distance for taking better photos.

Suppose the distance between a rectilinear lens and the shooting object is r_1, the distance between the lens and the image plane is r_2, and the dimension of the frame, which forms an image, is d (the film or image sensor). The lens is treated as if it is a pinhole (technically, the perspective center of a rectilinear lens is at the center of its entrance pupil) [9]. Then, the camera imaging process can be simplified as Fig. 4(a). To project a sharp image of distant objects, r_2 needs to be equal to the facial length f. Therefore, the AoV α can be calculated from the chosen dimension d and the effective focal length f as follows:

$$\alpha = 2 \arctan \frac{d}{2f} \cdot \frac{180}{\pi}, \tag{1}$$

where f represents 35mm equivalent focal length, and d represents the size of the film (or sensor). As we know, digital cameras often use different lenses and Charge-Coupled Devices (CCDs), and they have been mostly transformed into 35mm equivalent focal length and 35mm film (36mm wide and 24mm high) in metadata, so we can directly use them in HideMe.

Since pixels of the entire image and the targeted face are detected by face recognition, we could calculate the height of the sensed image by using digital zoom radio and face length (note that, each user could set his/her own face length



(a) (b)

Fig. 4. Photographic distance calculation (a): the camera imaging system, (b): image length calculation from a targeted face

manually). As shown in Fig. 4(b), the photographic distance can be calculated by trigonometric function as:

$$r_1 = \cot \frac{\alpha}{2} \cdot \frac{\pi}{180} \cdot \frac{Pix \cdot l}{2pix} \cdot z, \tag{2}$$

where Pix, pix, l and z represent pixels of the entire image, pixels of the targeted face, the length of face and digital zoom ratio, respectively. If there are multiple faces in the photo, we can use different lengths and pixels of faces to calculate different distances from each face to the camera.

E. Face Matching

Face matching aims to match a detected face from the sharing photo to a specific user in OSNs. The effective privacy policy enforcement depends on how successfully we can find the right person who appears in the shared photo. Since HideMe will be deployed in OSNs (e.g., Facebook) with massive amount of users, it is a challenge to efficiently match a detected face to a specific user. We propose a facial-attribute-based face matching algorithm, which can dramatically improve the efficiency of face matching, since we can use facial attributes to significantly reduce the number of users to be compared.

We use the state-of-art face recognition system, Tencent BestImage, to implement the face detection and recognition functions in HideMe [23]. Given a photo that is being shared, the faces will be detected and fed to the facial attributes classifier as shown in Fig. 5. Then, the facial attributes of each detected face will be predicted, which are used to reduce the number of users to be compared by finding users who have the identical facial attributes in the face database. The face database contains the identity of each registered user associated with pre-computed facial attributes and facial feature vector. Finally, we compare the feature vector of each detected

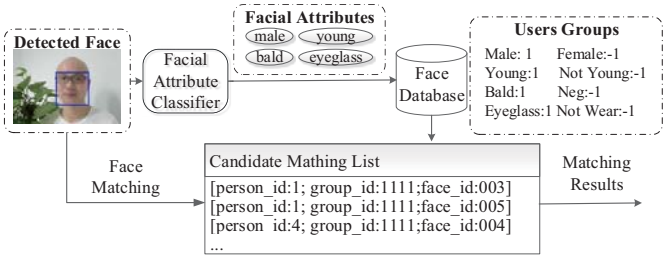


Fig. 5. Facial-attribute-based face matching algorithm

face with that of possible matched users. If the difference of two feature vectors is less than a predefined threshold, a match is identified.

Facial Attribute Classifier. To filter candidate faces based on facial attributes, we train a facial attribute classifier for facial attribute prediction based on the Adapted Balanced Convolutional Neural Network (ABCNN) [10] model, where a weighted objective function is constructed to improve the prediction accuracy.

Formally, we denote the set of input photos as $\mathbb{P}$, and the number of facial attributes which need to be predicted as N, respectively. When a photo $x \in \mathbb{P}$ is upload, the binary label of the photo's ith attribute can be presented as $y_i \in \{-1, +1\}$, where $i \in \{1, 2, \ldots, N\}$ represents the index of facial attributes. The $\mathbb{H}$ denotes possible decision functions in the hypothesis space, and $h_i(\theta^T x)$ represents the decision function, where $\theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$ is the network weights. Therefore, we define the loss function of the i-th facial attribute as $Loss_i(h_i(\theta^T x), y_i)$. $\mathbb{E}(Loss_i)$ is the expected loss over the range of inputs $\mathbb{P}$. Thus, the optimization task aims at the minimum expected squared error for each attribute.

$$\forall i : h_i = \arg\min_{h_i \in \mathbb{H}} \mathbb{E}(Loss_i). \tag{3}$$

Traditional approaches consider facial attributes as N independent tasks, and train independent classifier for each facial attribute, so it is hard to learn the latent correlations between attributes. To take advantage of these correlations, our classifier is trained to learn all these facial attributes simultaneously. In addition, the distribution of the attribute label should be identical between the training set and the testing set. Therefore, balancing the dataset can facilitate in training a better classifier.

In this work, we adopt ABCNN to train our facial attribute classifier by implementing weighted loss function to simulate a balanced dataset. Specifically, an adapted loss function is proposed by considering the distribution difference between training set and testing set as adapted weights. Firstly, we calculate the training distribution S_i for each attribute i, which uses the fraction of positive samples $Train_i^+$ ($0 < Train_i^+ < 1$) and negative samples $Train_i^-$ ($0 < Train_i^- < 1$) in the training set. An adapted weight is assigned for each class of attribute i by given binary testing distribution $Test_i^+$ and $Test_i^-$ (where $Test_i^+ + Test_i^- = 1$), as shown in Eq. (4) and Eq. (5).

$$w(i|+1) = 1 + \frac{Diff^+}{Test_i^+ + Train_i^+}, \tag{4}$$

$$w(i|-1) = 1 + \frac{Diff^-}{Test_i^- + Train_i^-}, \tag{5}$$

where $Diff^+ = Test_i^+ - Train_i^+$ and $Diff^- = Test_i^- - Train_i^-$. From the above equations, the weights can bring balance to the distribution difference between training set and testing set. If the fraction of positive or negative labels in the training set is less than the testing set, the weight of the i-th facial attribute will be increased. Similarly, the weight of positive or negative labels in the training data will be decreased if it is higher than those in the testing data. Then, these adapted weights are integrated into the mixed loss function. As a result, a weighted mixed task square error is adopted as the loss function instead of regular hinge-loss function, and the optimization problem of an M-element training set X with labels Y can be expressed as:

$$\forall i : \arg\min_{h_i \in \mathbb{H}} \mathbb{E}(L(X, Y)) = \arg\min_{h_i \in \mathbb{H}} \mathbb{E}(\sum_{j=1}^{M} \sum_{i=1}^{N} w(i|Y_j i(x))||h_i(X_j) - Y_j i||^2). \tag{6}$$

We can implement the facial attribute classifier based on ABCNN model by replacing the standard loss layer of a Deep Convolution Neural Network (DCNN) with a layer implementing Eq. (6).

V. Implementation Issues

HideMe is available on Github[7]. It depends on a MySQL database server, which stores the uploaded photos, users' privacy policies and other information. The social relation information is stored in a neo4j database server.

Installation. When a user installs HideMe, he needs to register his names and face with the server, upload some of his photos, and add more personal information such as face length (18cm as a default). The user interface of HideMe is shown as Fig. 6.

Face Processing. Once a photo is uploaded, the detected faces are marked and the users' names are displayed as shown in Fig. 6(a). We utilize the online API service of Tencent BestImage[8] to implement the face detection and recognition in HideMe. To predict facial attributes, we implement the proposed ABCNN network by replacing the final loss layer of a VGG-16 network [18] with the loss function in Eq. (6). Since we fix the input dimension of the ABCNN network as 128×128, each detected face will be scaled to the same size before being fed into this classifier.

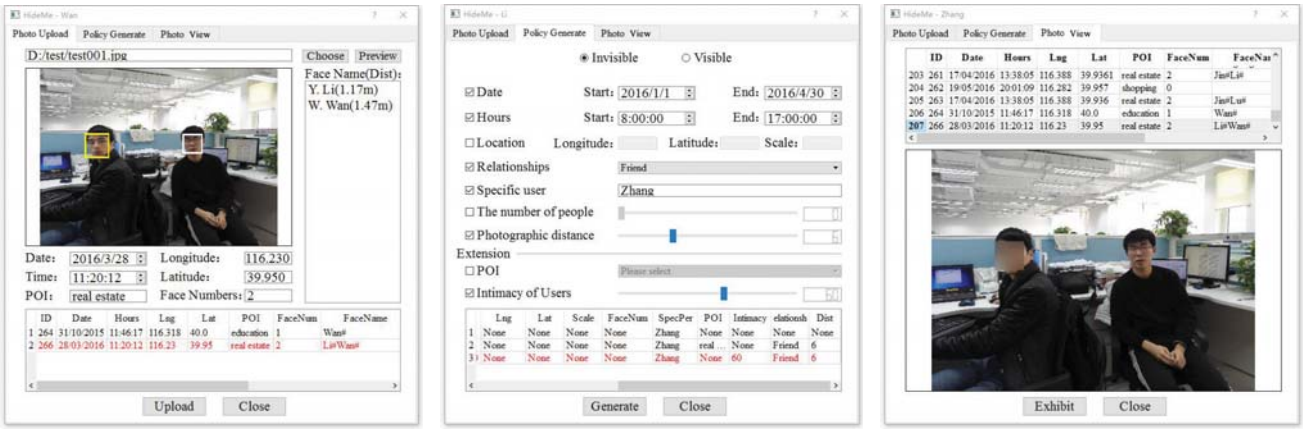
Reverse Geo-coding. PoIs have been widely used in location based services. In our work, we obtain such information from the latitude and longitude by using Reverse Geo-coding API by Baidu Map[9], Google Map[10].

[7]https://github.com/HideMe2018/HideMe

[8]https://bestimage.qq.com/

[9]http://lbsyun.baidu.com/index.php?title=webapi/guide/webservice-geocoding

[10]https://developers.google.com/places/web-service/



(a) Photo Uploading (b) Policy generating (c) Photo viewing

Fig. 6. User interfaces of HideMe

Intimacy of Users. Since some users may want to use an exact value to represent how close their relationship of photo-viewer is, we employ the Jaccard's Index, which is a common measurement on social strength, to calculate the intimacy. Note that it can be replaced by other methods. The function is given as follow:

$$J_{ij} = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}, \tag{7}$$

where $\Gamma(i)$ and $\Gamma(j)$ are the friend sets of node i and node j. The implication of Jaccard's index is the ratio of common friends and total friends of node i and node j.

VI. Performance Evaluations

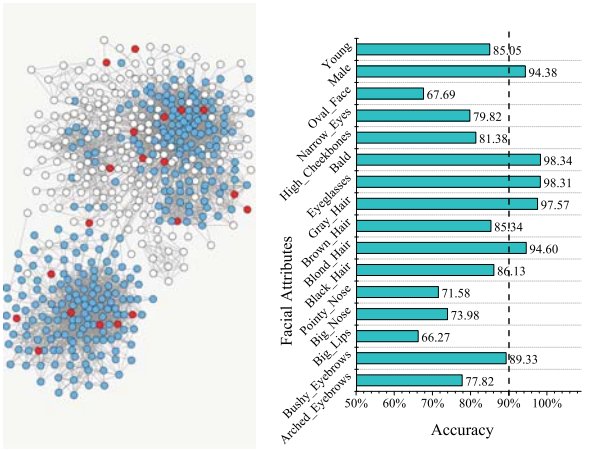
A. DataSet

OSN dataset for Photo Sharing. HideMe is run on a near real OSN, as shown in Fig. 7(a). We use a public Facebook dataset with 4,039 nodes and 88,234 edges provided by Stanford SNAP[11] to design a basic social relationship network. Then, we collect 2,000 photos from 20 volunteers (red nodes) and their friends (blue nodes), and such photos are all taken by smartphones in a consecutive time. White nodes represent the friends of friends. Next, we add pseudonyms and faces information for the collected users to build the photo sharing dataset.

Attribute Set for Face Filtering. We adopt the CelebA dataset to train the ABCNN network of Facial-attribute Classifier. The dataset consists of about 20 images for each of 10,177 celebrities, and there are 202,599 images in total. In the CelebA dataset, 70% images are used as training set, 20% images are used for validation and the remaining 10% images for testing. Each image in the CelebA dataset is annotated with binary labels of 40 facial attributes. We pick 16 out of the 40 attributes that do not change frequently for the same person as our candidate attributes for filtering.

We evaluate the classification accuracy of each candidate facial attributes shown in Fig. 7(b). The average accuracy over those 16 attributes is 88.53%. Out of the 16 facial attributes, we choose {Male, Young, Eyeglasses, Bald} as our attributes set for face filtering, due to their high accuracy and stability for the same person. The classification accuracy of {Male, Young, Eyeglasses, Bald} achieves 94.37%, 85.04%, 98.31% and 98.34% respectively.

[11]http://snap.stanford.edu/data/



(a) (b)

Fig. 7. Dataset (a): our photo-sharing OSN (500 Nodes), (b): accuracy of each facial attribute on CelebA

B. Evaluations of HideMe

1) Cost Analysis: We run HideMe on a Thinkpad T430u laptop with Intel Core i7-3517U and 16G RAM. To test user experience in HideMe, we analyze the time consumptions of the photo uploading and photo viewing.

Photo Uploading. We randomly pick 100 photos from our volunteers' photos and process them with HideMe. The whole phase (without face recognition) takes 7.5824ms on average per photo. It is a little longer than the 2.3ms shown in Face/Off [8], since HideMe has to extract and compute photo factors. Specifically, it takes 7.4763ms for Metadata Extraction and 0.0035ms for Photographic Distance, respectively.

Photo Viewing. We randomly access 100 photos which have no permission to view some faces, and measure the overhead in worst-case scenarios. As shown in Fig. 8, it costs 73ms when we put the blur function in the viewing phase, 52ms when we put the function before (similar as [8]). Although the

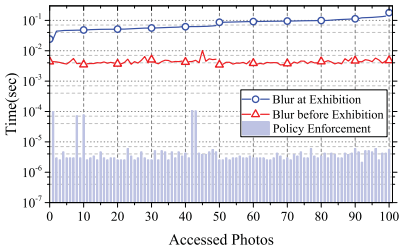
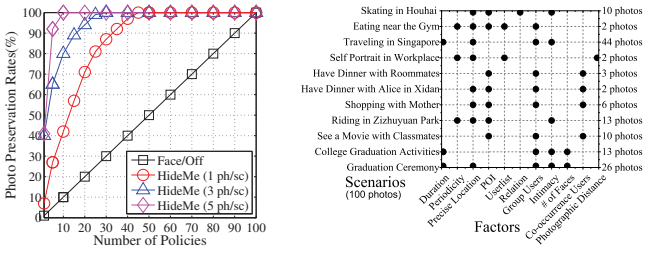


Fig. 8. The total time required for a photo



(a) (b)

Fig. 9. Policy efficiency evaluations (a): the ratio of photos to policies - photo preservation rate, (b): the example of policy generating

second approach is time efficient, it consumes more storage space for storing the image block of each blurred face, which is much larger than the metadata and the tag, especially for large scale scenarios.

Scalability. We measure the impacts of additional faces and policies on the scalability. We select the testing photos from another 100 random photos, where each photo has more than 3 faces and each face has more than 5 relative policies. Each extra policy costs about 0.008ms as shown in Fig. 8.

2) A comparison: To demonstrate the effectiveness of HideMe, we asked volunteers to assign policies for 100 photos using Face/Off and HideMe, respectively. The results show that HideMe outperforms Face/Off. We also found that the policy amount is closely related to the number of scenarios in the samples. Fig. 9(a) shows three typical volunteers' shared photos, which have 1, 3, and 5 photos per scenario. It is easy to see that HideMe provides higher photo preservation rate for the shared photos by utilizing a smaller amount of policies. We also demonstrate how to build scenarios in Fig. 9(b), which implies that photo preservation rate can be improved by using more abstract polices.

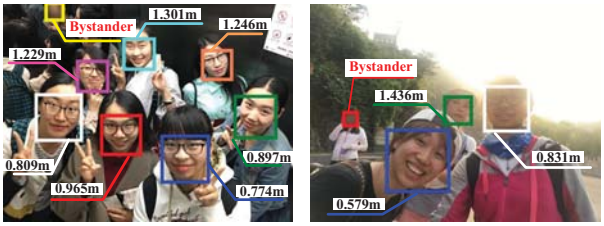
C. Evaluation Results of Distance-based Bystander Detection

According to the study results in Sec. III-A, it is hard for a user to realize that he/she becomes a bystander outside the distance of 4m from the camera lens. Thus, we recommend 4m as the threshold of photographic distance in default.

To evaluate the effectiveness of our Photographic Distance calculation, We take some photos from different distances as 1.2m, 2.4m, 3.6m and 4.8m. Since the digital zoom ratio may affect the detection, we conduct a set of tests in different digital zoom ratios as 1, 2.75 and 4. Clearly in Tab. IV, there is a trend that the bigger digital zoom ratio leads to higher deviation. These is also a deviation of the first (0.809m) and the second

TABLE IV
TEST RESULT OF PHOTOGRAPHIC DISTANCE

The actual distance	1.20m	2.40m	3.60m	4.80m
digital zoom ratio = 1	1.21m	2.35m	3.75m	4.69m
deviations	-0.01m	0.05m	-0.15m	0.11m
digital zoom ratio = 2.75	1.14m	2.42m	3.46m	4.39m
deviations	0.06m	-0.02m	0.14m	0.41m
digital zoom ratio = 4	1.12m	2.19m	3.36m	4.76m
deviations	0.08m	0.21m	0.24m	0.04m



(a) (b)

Fig. 10. The example for bystander detection (a): indoor, (b): outdoor

girl (0.965m) on the left side of Fig. 10(a). However, all deviations of photographic distance in our experiment are acceptable for detecting bystanders.

D. Facial-attribute-based Face Matching

Effectiveness of Face Filtering. To evaluate the effectiveness of face filtering, we randomly pick 1,000 people from the CelebA dataset, and perform face filtering on a face image of each person. The face matching range can dramatically decrease to 2,071 persons on average, which is only 20% of all 10,177 people. In addition, 95% of selected 1,000 face images can be matched in the shortlist.

Performance of Face Filtering. To evaluate the performance of face filtering, we split the face database into two parts. The first part stores all face images of each person and is labelled as 'group all'. In the second part, we name the group by the values of the selected facial attributes. For instance, given a face image with facial attributes {Male=-1, Young=1, Eyeglasses=1, Bald=-1}, this image will be put into the group named by '-111-1' associated with its 'person id'. Since there are four facial attributes for face filtering, and each attribute is a binary value, we have 16 groups in the second part. In this way, we can make face matching with face filtering in the second part. To evaluate the performance and scalability of face filtering, we compare the performance of face matching using face filtering with that of matching face in the whole database by changing the size of the database. We randomly select 100 persons from the CelebA dataset, and perform face matching on a face image of each person. In Tab. V, the result shows the face filtering based face matching is more efficient than performing face matching in the whole database. Furthermore, with the increasing size of the database, the face filtering based face matching demonstrates its higher scalability in a large scale database than directly matching face in the whole database.

Accuracy of Facial-attribute-based Face Matching. We compare the accuracy of the proposed facial-attribute-based face matching with the state of art face recognition system

TABLE V
SCALABILITY EVALUATION OF FACE FILTERING (UNIT: MS)

Group	The number of persons in the database				
	1000	2000	4000	8000	10000
'group all'	223.57	250.33	296.86	353.98	424.89
facial attributes	193.08	197.63	200.68	226.94	246.42

TABLE VI
THE ACCURACY COMPARISON: FACIAL-ATTRIBUTE-BASED FACE MATCHING V.S. TENCENT BESTIMAGE

Method	Database: 4000 persons		Database: 10000 persons	
	Time Cost	Accuracy	Time Cost	Accuracy
Facial-attribute-based Face Matching	200.68 ms	95.3%	246.42 ms	94.7%
Tencent BestImage	296.86 ms	97.3%	424.89 ms	96.7%

Tencent BestImage, which adopts kNN approximate face matching based on deep features. In this experiment, we set $k = 5$ in the Tencent BestImage face recognition system. We randomly pick 150 people from CelebA dataset, and perform face matching on a face image of each person within two databases of 4,000 and 10,000 persons from CelebA dataset, respectively. These two databases contain the 150 tested people. In Tab. VI, the results shows the proposed facial-attribute-based face matching is much more efficient with only a tiny accuracy degradation compared with Tencent BestImage. Note that as the database size increases, for example in real applications such as Facebook, saving almost half of the running time will be significant.

VII. CONCLUSIONS

In this paper, we designed, implemented and evaluated a privacy-preserving photo sharing framework, called HideMe, which could help associated friends preserve their privacy in different scenarios in online photo sharing. HideMe extracts important factors when users upload their photos. It provides the associated friends one-time settings based on a scenario-based access control model, instead of setting policies photo-by-photo. As a result, HideMe helps users hide their faces on related photos from unwanted viewers by one time policy generation. We also designed a distance-based algorithm to identify and protect the privacy of bystanders. Moreover, HideMe not only protects users' privacy but also reduces the system overhead by a carefully designed face matching algorithm. Evaluation results demonstrated its effectiveness.

ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (61672515, 61872441), the National Key R&D Program of China (2017YFB0802203) and the Youth Innovation Promotion Association CAS.

REFERENCES

[1] Abokhodair, N., Hodges, A., Vieweg, S.: Photo sharing in the arab gulf: Expressing the collective and autonomous selves. In: Proc. of ACM CSCW 2017
[2] Aditya, P., Sen, R., Druschel, P., Oh, S.J., Benenson, R., Fritz, M., Schiele, B., Bhattacharjee, B., Wu, T.T.: I-pic: A platform for privacy-compliant image capture. In: Proc. of ACM MobiSys 2016
[3] Aronov, B., Efrat, A., Li, M., Gao, J., Mitchell, J.S., Polishchuk, V., Wang, B., Quan, H., Ding, J.: Are friends of my friends too social? limitations of location privacy in a socially-connected world. In: Proc. of ACM MobiHoc 2018
[4] Fogues, R.L., Murukannaiah, P.K., Such, J.M., Singh, M.P.: Sharing policies in multiuser privacy scenarios: Incorporating context, preferences, and arguments in decision making. ACM Transactions on Computer-Human Interaction 24(1), 5 (2017)
[5] Griffin, P.F.: The correlation of english and journalism. The English Journal 38(4), 189–194 (1949)
[6] Guo, Y., Yin, L., Liu, L., Fang, B.: Utility-based cooperative decision in cooperative authentication. In: Proc. of IEEE INFOCOM 2014
[7] Hu, H., Ahn, G.J., Jorgensen, J.: Multiparty access control for online social networks: model and mechanisms. IEEE Transactions on Knowledge and Data Engineering 25(7), 1614–1627 (2013)
[8] Ilia, P., Polakis, I., Athanasopoulos, E., Maggi, F., Ioannidis, S.: Face/off: Preventing privacy leakage from photos in social networks. In: Proc. of ACM CCS 2015
[9] Kerr, D.A.: The proper pivot point for panoramic photography. The Pumpkin 2(8), 1–15 (2008)
[10] Li, A., Du, W., Li, Q.: Politecamera: Respecting strangers' privacy in mobile photographing. In: Proc. of EAI SecureComm 2018
[11] Li, A., Li, Q., Gao, W.: Privacycamera: Cooperative privacy-aware photographing with mobile phones. In: Proc. of IEEE SECON 2016
[12] Li, F., Li, H., Jia, Y., Yu, N., Weng, J.: Privacy computing: concept, connotation and its research trend. Journal on Communications 37(4), 1–11 (2016)
[13] Li, F., Li, Z., Han, W., Wu, T., Chen, L., Guo, Y., Chen, J.: Cyberspace-oriented access control: A cyberspace characteristics based model and its policies. IEEE Internet of Things Journal (to appear)
[14] Li, F., Sun, Z., Niu, B., Guo, Y., Liu, Z.: Srim scheme: An impression-management scheme for privacy-aware photo-sharing users. Engineering 4(1), 85–93 (2018)
[15] Olejnik, K., Dacosta, I., Machado, J.S., Huguenin, K., Khan, M.E., Hubaux, J.P.: Smarper: Context-aware and automatic runtime-permissions for mobile devices. In: Proc. of IEEE SP 2017
[16] Pallas, F., Ulbricht, M.R., Jaume-Palasí, L., Höppner, U.: Offlinetags: A novel privacy approach to online photo sharing. In: Proc. of ACM CHI 2014
[17] Roesner, F., Molnar, D., Moshchuk, A., Kohno, T., Wang, H.J.: World-driven access control for continuous sensing. In: Proc. of ACM CCS 2014
[18] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
[19] Such, J.M., Criado, N.: Resolving multi-party privacy conflicts in social media. IEEE Transactions on Knowledge and Data Engineering 28(7), 1851–1863 (2016)
[20] Such, J.M., Porter, J., Preibusch, S., Joinson, A.: Photo privacy conflicts in social media: A large-scale empirical study. In: Proc. of ACM CHI 2017
[21] Toubiana, V., Verdot, V., Christophe, B., Boussard, M.: Photo-tape: user privacy preferences in photo tagging. In: Proc. of ACM WWW 2012
[22] Vishwamitra, N., Li, Y., Wang, K., Hu, H., Caine, K., Ahn, G.J.: Towards pii-based multiparty access control for photo sharing in online social networks. In: Proc. of ACM SACMAT 2017
[23] Wang, D., Otto, C., Jain, A.K.: Face search at scale: 80 million gallery. In: Proc. of IEEE ICB 2015
[24] Wang, D., Otto, C., Jain, A.K.: Face search at scale. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), 1122–1136 (2017)
[25] Xu, K., Guo, Y., Guo, L., Fang, Y., Li, X.: My privacy my decision: Control of photo sharing on online social networks. IEEE Transactions on Dependable and Secure Computing 14(2), 199–210 (2017)
[26] Xu, Y., Price, T., Frahm, J.M., Monrose, F.: Virtual u: Defeating face liveness detection by building virtual models from your public photos. In: Proc. of USENIX Security 2016
[27] Yin, L., Guo, Y., Zhang, H., Huang, W., Fang, B.: Threat-based declassification and endorsement for mobile computing. Chinese Journal of Electronics (to appear)
[28] Yus, R., Pappachan, P., Das, P.K., Mena, E., Joshi, A., Finin, T.: Faceblock: privacy-aware pictures for google glass. In: Proc. of ACM MobiSys 2014
[29] Zhang, L., Jung, T., Liu, K., Li, X.Y., Ding, X., Gu, J., Liu, Y.: Pic: Enable large-scale privacy preserving content-based image search on cloud. IEEE Transactions on Parallel and Distributed Systems 28(11), 3258–3271 (2017)
[30] Zhang, L., Liu, K., Li, X.Y., Liu, C., Ding, X., Liu, Y.: Privacy-friendly photo capturing and sharing system. In: Proc. of ACM UbiComp 2016