Privacy.Tag: Privacy Concern Expressed and Respected

Cheng Bo[1,2,5], Guobin Shen[2], Jie Liu[3], Xiang-Yang Li[4,5], YongGuang Zhang[2], Feng Zhao[2]

[1]Department of Computer Science, University of North Carolina at Charlotte, USA
[2]Microsoft Research, Beijing, China [3]Microsoft Research, Redmond, USA
[4]School of Software and TNLIST, Tsinghua University, China
[5]Department of Computer Science, Illinois Institute of Technology, USA

cbo1@uncc.edu, Jacky.Shen@microsoft.com, liuj@microsoft.com, xli@cs.iit.edu, ygz@microsoft.com, zhao@microsoft.com

Abstract

The ever increasing popularity of social networks and the ever easier photo taking and sharing experience have led to unprecedented concerns on privacy infringement. Inspired by the fact that the Robot Exclusion Protocol, which regulates web crawlers' behavior according a per-site deployed robots.txt, and cooperative practices of major search service providers, have contributed to a healthy web search industry, in this paper, we propose Privacy Expressing and Respecting Protocol (PERP) that consists of a Privacy.tag – a physical tag that enables a user to explicitly and flexibly express their privacy deal, and Privacy Respecting Sharing Protocol (PRSP) – a protocol that empowers the photo service provider to exert privacy protection following users' policy expressions, to mitigate the public's privacy concern, and ultimately create a healthy photo-sharing ecosystem in the long run. We further design an exemplar Privacy.Tag using customized yet compatible QR-code, and implement the Protocol and study the technical feasibility of our proposal. Our evaluation results confirm that PERP and PRSP are indeed feasible and incur negligible computation overhead.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Design, Measurement, Security

Keywords

Privacy protection, Photo sharing, QR-code

SenSys'14, November 3–5, 2014, Memphis, TN, USA.
Copyright 2014 ACM 978-1-4503-3143-2/14/11 ...$15.00
http://dx.doi.org/10.1145/2668332.2668339

1 Introduction

Taking and sharing photos have become easier with the proliferation of devices with cameras, high-bandwidth mobile networks, and photo-service providers (PSP), such as social networking sites, photo-sharing applications and portals. The latest reports show that approximately 1.4 million photos are uploaded to Flickr every day, and the number reaches an astonishing 40 million for Instagram [8, 9]. The trend is accelerating with emerging wearable devices such as Google Glass [5] and Memoto [11], and Kinect in the home. In the case of Google Glass, pictures are shared in real time on Google+ and, depending on user settings, automatically spread through the user's social network, without human in the loop.

The ease of taking and sharing photos, along with new features on PSPs for face recognition, automatic tagging, and linking to one's online profiles, have triggered an outcry of concerns about privacy from the public. For example, on 17th May 2013, shortly after Google Glass was announced, eight members of US Congressional Bi-Partisan Privacy Caucus sent a letter to Google seeking answers about Glass's privacy implications [2]. Ironically, even Google banned Google Glass at their own events [4].

Currently, photo and video privacy control solutions put a cognitive burden on the subjects being photographed. Most countries dictate that cameras and video recorders must make sound and visual cues to show that recording is in action, to give the subject an opportunity to adjust his/her behavior or speech accordingly. When a person finds his/her photo is unwillingly shared on a site (e.g. Facebook), he/she may request to be de-tagged. However, with the upcoming surge of mobile and wearable devices, it is difficult for a person to keep track of which nearby devices maybe recording, or where the photos will end up at. Research has also shown that users struggle to configure their photo sharing policies correctly [33]. Do people have to relinquish their privacy in the era of mobile and wearable computing?

In this paper, we propose a paradigm to *return privacy control to people* being photographed. Here, we make an analogy to digital information on the web. We view people as the owners of their information, images, behaviors, speech, etc. as contents. Sensors, including cameras, are *crawlers* in the physical world. They digitize physical information and send them online, sometimes automatically. PSPs and various online channels, are services that organize, index, and serve this information to other users. We are further inspired by the Robot Exclusion Protocol (REP), also known as robots.txt, that expresses owners' desires about how the information should be used by crawlers and online services. The desires are then respected by online service providers to create a healthy web ecosystem.

Our proposed solution, the *Privacy Expressing and Respecting Protocol* (PERP) is more of an architecture gateway, which consists of two parts: 1) a privacy expression tag, (called Privacy.Tag, or Tag for short) and 2) a privacy respecting sharing protocol (PRSP, or Protocol). A Privacy.Tag, worn by a person, is an imagery equivalent to robots.txt. The Tag expresses the privacy desires of the person, such as whether he/she wants the photo to be shared, any restrictions on where to share/whom to share to, and how he/she wants his/her images to be protected to avoid unauthorized people to see or use these photos. Notice that such privacy desires are not necessarily encoded in the tag. The tag could just encode a link where such privacy desires are encoded and stored. The PRSP, on the other hand, empowers well-behaved PSPs to process the pictures and block parts of the images that are from subjects who wear Tags to respect their privacy expressions.

The goal of PERP is to promote a healthy photo sharing ecosystem so that people can feel at ease around wearable camera devices, and reduce the burden for PSPs to process requests from subjects to implement "right to be forgotten" aftermath [41]. We emphasize up front that PERP is *also* a privacy protection/enforcement solution. In fact, it does not provide anything to protect subjects from privacy attacks, except possibly providing an easy way to scan web images for PERP violations. Rather, it embodies more a notion of usable privacy, and provides a mechanism, by which, when followed by PSPs, explicitly expressed privacy appeals will be automatically protected. This is the same way in which robots.txt does not prevent malicious web crawlers from retrieving and exposing web contents but a search engine or other services that respect the desire will gain its reputation and thus draw more traffic. We hope similarly that our protocol and mechanism will help PSPs to gain advantages over the long run by respecting the privacy protection protocols.

Our technical focus is on proving the feasibility of PERP through system design, implementation, and evaluation. We have tackled the following challenges:

The first challenge is the tag design, which should be reliably detectable yet less noticeable. They have to be localized to pinpoint the wearers and work with all cameras including legacy ones. They must either contain the privacy policies directly or point to where the policies are. We believe different types of Tags can be designed. In our particular design, we have chosen to use QR-codes but customize them by embedding a color-reversed position locator in the center of the QR-code to ensure reliable detection while avoiding confusion with other existing QR-codes.

The second challenge is on how to grant the privacy control, i.e., the control of photos' publicity scopes back the user. To this end, we design a reversible, pattern guided obfuscation process to protect the face. The protector generates a random pattern for a photo to serve as a *protection key*, and shuffle blocks in the face area according to the protection key. The original face can be restored by shuffle blocks back when the protection key is known. The protection key is encrypted with the targeted user's public key retrieved from the Tag worn by this user, or his own public key, depending on whether the Tag is decodable. The encrypted public key is then embedded into the picture's header as part of the annotation of a processed Tag. Any user with the corresponding private key can obtain the protection key and restore the original photo. The user can thus control the scope of a photo's publicity by controlling the dissemination of his/her private key, no matter who took and shared the photo. This design also saves the PSP from storing the original copies. For example, when providing evidence for law enforcement purposes the original photo shall be presented despite the criminal has worn a Privacy.Tag.

The third challenge is matching a Tag worn by a user to the correct face in the photo (or any other parts of the user that may need privacy protection). Given the extensive study on face detection and recognition in computer vision, there is surprisingly little work on body detection. Without depth information, body detection is extremely hard as users wear various clothes and vary their postures. In this work, we have developed a heuristic, range-constrained face/Tag matching algorithm, based on the assumption that the Tag is worn on the upper body. We leverage the size, position and orientation of a detected face to constrain the possible range of a Tag, and find the closest one if multiple Tags are detected.

We have implemented and evaluated various aspects of our design in realistic settings. The evaluation results confirm that: 1) our Tag design is effective, and a 5cm-10cm sized Tag can effectively express user's privacy appeal in most situations; 2) the proposed face/Tag matching algorithm works reasonably well, achieving 96% precision and 77% recall rate for indoor office, and 77% precision and a 78% recall rate for outdoor park environments; 3) the computational overhead is mainly on the face detection, which takes up to 95.6% of CPU time. As many PSPs have already rolled out automatic face detection and tagging services, the additional overhead attributed to the proposed PRSP is negligible. The overhead of removing protection for authorized viewers is of the order tens of milliseconds. In summary, our experimental evaluation confirms the feasibility of the proposed Privacy.Tag and the associated Privacy Respecting Protocol.

The main contribution of our work is the proposal of a new privacy protection paradigm that aims to give the privacy control back to people being photographed. We propose a Privacy Expressing and Respecting Protocol (PERP) and a Privacy.Tag concept. We also analyze the whole lifecycle of a photo and identify that the PSPs are the best places to exert privacy protection. At the current stage, our work is a framework and proof of concept rather than a full-blown system. There are multiple technical challenges as discussed near the end of the paper. We hope to raise people's awareness of the possibility of a tangible privacy solution, trigger more researchers to come up with better mechanism design, and advocate mainstream PSPs to embrace such a solution.

2 Photo Privacy: Practices and Challenges

The current practice of imagery privacy policies are extension of policies that are designed to protect personal information, building upon three notions: disclose, consent, and damage control.

Disclose. Disclose means that people or organizations who collect, store, or share pictures need to disclose their practice. Forms of disclosure can be indicators on camera devices that show their activities and privacy statements on PSP websites. For example, South Korea mandates 64 decibel shutter sound by law since 2004 [14], and Japan compels device manufactures to utter a shutter sound when taking photos [10]. A bill of "Camera Phone Predator Alert Act" [1] was also proposed in 2009 in US for the same purpose.

Consent. In addition to disclosing their practices, some organizations allow subjects to control what information can be collected and how they can be shared. By default, the user can either opt in (i.e. all information is collected) or opt out (i.e. nothing is collected). Most of the time, a user must opt in to use the service. Studies have also shown that most users have difficulties in configuring PSPs' privacy settings. Users' actual privacy settings are usually inconsistent with their sharing intentions [33]. For photo sharing, the current social norm is that the subjects are opted in by default.

Damage Control. Personal data can be leaked through social media and cause damage [34,40]. When private information is compromised, some online service providers allow users to contest and will take steps to control the damage by un-tagging people, removing pictures, or deleting online history. This process is typically manual and cumbersome.

It is worth noticing that photo taking and sharing are fundamentally different from other personal information collection because it involves two parties – the photographer (broadly defined to refer to anyone who take pictures) and the subjects being photographed. In today's practice, disclose and consent only applies to the photographer. In many cases, a subject does not know ahead of time what pictures are taken and where they are posted. So, damage control becomes the only defense aftermath.

In everyday life, photo privacy are typically achieved through direct human involvement, such as posting signs at the entrances of locker rooms showing that "cell phones are not allowed." "Stop the Cyborgs" [15] tries to shape social norms and ask, by way of special posters (*e.g.* 'Google Glass Ban Signs'), people to remove their wearable devices in social or private contexts. TagMeNot uses special tags to let people express their privacy concern and calls for phototakers to deliberately avoid taking photo of them [16]. Some online service providers voluntarily take steps to protect imagery privacy. For example, Bing StreetSide and Google Street View [6] blur all faces and license plates [7] in the images before serving them publicly. However such a blurring-all solution is obviously not viable for photos sharing scenarios.

Challenges: The goal of PRSP is to allow potential photo subjects to proactively express their privacy preference and to promote a healthy privacy-respected photo sharing ecosystem. We choose wearable tags for their flexibility and widely available tool chains. However, a practical system must address the following challenges.

Reliability. The wearable tag should be reliably detectable yet not very intrusive to wear. They need to be sharply localizable to pinpoint the wearer and should work with all cameras including legacy ones, in addition to certain information-carrying capability to embed user's privacy policy. The detected tags must be reliably matched to the right faces.

Flexibility. People's privacy desire may be diverse and often situation dependent. The users themselves should be empowered to control the publicity scope of their photos, in addition to flexibly express their privacy desires. In some cases, a subject may want to recover the original images afterward for controlled sharing. Or the law enforcement may request original images under warrant for crime investigations. So, the privacy protection mechanism should allow the process to be reserved.

3 PERP: Concepts and Design Overview

We address the above challenges through the design of Privacy.Tags, associated Tag recognition and inversable face protection algorithms, and a privacy-respecting sharing protocol (PRSP, or simply Protocol).

3.1 The Concept

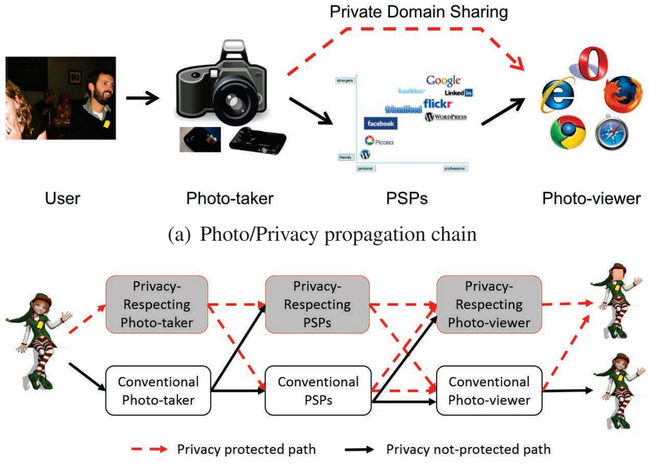
The concept of Privacy.Tag (or simply Tag) is to design a special wearable tag to let a user explicitly express her desire of privacy by wearing a Tag, and convey user specified privacy policies in the Tag. The PRSP is set up to empower major PSPs to respect those explicitly expressed privacy appeals. Other players of the photo sharing ecosystem, *e.g.* device manufacturers and/or photo-sharing App developers, and even browsers, are encouraged to respect the Protocol as well.

The Privacy.Tag and the PRSP are inspired by the use of *robots.txt* [18] to specify the allowed or disallowed contents on websites and the Robot Exclusion Protocol (REP) [37] to regulate web crawlers' behaviors. Although some dishonest crawlers may ignore it, major crawlers, especially those from search giants, all respect REP and discipline their crawling behaviors accordingly. The collective rational of major players leads to the healthy web search industry we see today.

3.2 Design Considerations

Diverse Privacy Appeals: In real life, different users have different privacy appeals; even for the same person, the privacy desire may be situation dependent. We empirically classify privacy appeals regarding photo-sharing into three general categories:

- Absolute privacy: Photos should not be publicized, and always protected if they are indeed shared. This typically happens in very private situations.
- Controlled publicity: Photos may be taken and shared within certain publicity scope controlled by the user. Photos outside the scope should be protected. This commonly happens at social gatherings.



(a) Photo/Privacy propagation chain

(b) Impact of privacy protection at different stages

Figure 1. Photo and privacy propagation chain and impact of privacy protection at different stages.

- Full publicity: Photos can, or even should, be published without protection. This is usually the case when attending public events.

Diverse privacy appeals call for a way that is flexible in expressing the privacy appeal and can effectively control the scope of publicity. The privacy control should be granted back to users themselves.

PSPs Being the Narrow Waist: Figure 1-(a) depicts a typical flow of photo sharing: a photo is first captured by a photo-taker with a camera, a conventional Point&Shoot camera, a DSLR or a camera on a mobile or wearable device. It may be shared either via private sharing channels(e.g., through emails) or to the public-facing PSPs. Finally, the pictures reach photo-viewers through various web browsers or Apps.

While it is true that if any players in the photo sharing chain exert the Protocol, the users' privacy expression can be respected, we believe PSPs are the narrow waist to implement it. First of all, there are many camera enabled devices, some with very little computing resources to do tag detection and face recognition. When a photo is shared to the public, then the privacy of people being photographed in the picture is at risk. However, people can take pictures at their own will. When someone takes a photo without sharing it online, it is hard to say that the photo sharing policy of the subject is violated. Therefore, we take the view in this paper that as long as a photo is not shared publically without permission, there is no privacy leakage issues. On the image display side, once the unprocessed images left the servers, user privacy are open to be infringed. Finally, there are only a few popular PSPs that dominate the photo sharing services and they have well-defined API for photo upload. They also have already integrated features like face recognition into the photo upload chain. While we focus on PSPs in this paper, we acknowledge that the Protocol should be recommended to all players in the ecosystem, especially the upper stream device manufacturers and photo-sharing App builders. Developing privacy-policy-respecting camera would be the most effective way to protect people's privacy. However, given the large amount of cameras already in use and the wide penetration and huge diversity of new camera-equipped mobile devices, deliberately posing any assumption on the camera is not feasible. Thus we go after a cooperative model between PSPs and users' privacy desire.

Possibility of Establishing PERP: Wide spread public concerns on privacy clearly endorse the desire for users to express and request privacy protection. We believe that mainstream PSPs are rational players of a big ecosystem. They may also have the motivation or at least is willing to respect others' privacy, especially when a welcomed trend or obligatory regulations are in place. As aforementioned, all PSPs have already put certain privacy policies in place, albeit they may not be effective.

Our promotion of PERP is also encouraged by a recent regulation, Do-Not-Track [39], launched by US FTC to regulate targeted advertising not to reveal users' behaviors or profiles to ad networks. All major Internet browsers have implemented the Do-Not-Track feature by now.

Finally, we argue that wearable and mobile device manufactures should exert privacy protection not only for its maximal effectiveness but also for an economic incentive. For instance, privacy-respecting wearable devices could be more welcomed by users, or at least face less risk of being protested.

3.3 Privacy Policy

A privacy policy specifies the allowed publicity scope when one's photos are shared online, and also establishes a handle for the user to gain the control of publicity. As the policy, whole or partial, needs to be embedded in a Privacy.Tag, we need to balance the compactness and the flexibility of policies. Our design is as follows:

```
PK: user's personal public key
+: allowed domains, or * for all
-: disallowed domains
url: privacy_policy_site/UID/#n
```

Wearing a Tag is already a sign of privacy. Hence the default behavior of PRSP is always to protect the privacy upon Tag detection. Different sites have very different ways to allow their users to configure and control privacy policies. A person may only want the photo to be shared on a PSP that she is on and where she understands and has configured her privacy policies. She can achieve this by turning off default on protection that PSP. She does this using a whitelist via the `'+:'` syntax. The user may use a `'*'` to allow no protection for all sites. Note that, however, the effect of wearing a `'+: *'` Tag is different from not wearing a Tag, for the cases when the Tag is detected but cannot be decoded. For flexibility, we also allow a user to explicitly specify disallowed domains via `'-:'` syntax. Multiple allowed or disallowed domains may be specified. Each domain takes one line. We impose a rule of ordering: in cases of overlapping domain names, top ones always overwrite bottom ones.

The public key is used by privacy protector (*e.g.* a PSP) to encode the secret protection key. Legitimate users holding the corresponding private key can thus decipher the secret protection key and restore the original photo. The user can control the publicity scope by controlling the distribution of the private key using methods such as Attribute Based Encryption [24, 29]. Not specifying a public key implies the user cannot revoke the protection, nor will anyone outside the allowed sites.

We also include a `url` field to redirect PSPs to the full list of one's privacy policy residing on dedicated web sites that anonymously host users' privacy policies.[1] Although embedding `urls` with all user's privacy requirements into the QR-code alone may simplify the QR-code design, we still encourage users to specify a partial lists of the policy in the tag. By doing this, the privacy preserving process could be accomplished locally according to the PSPs' white or black list carried in the tag before the photo is published through certain PSP API, so as to reduce the burden on both PSPs and the policy hosting site. Each user can apply a page there to express their customized private policies. Multiple policies can be specified and indicated with `#n`. Assume the page is indexed by a 16-byte unique ID. Shortened URLs can be applied. Note that, all the fields are optional. When none of them appears, it implies protection for all sites.

3.4 PERP Design

The proposed PERP consists of simple rules for users and for PSPs.

On The User Side: A user specifies her own privacy policy. The policy completely embedded to a Privacy.Tag if the policy is short, or be a URL pointing to a web page that hosts the details. When she wants to express her privacy appeal, she wears the corresponding Tag.

On The PSP Side: All PSPs (and optionally other players in the ecosystem) will perform the Privacy.Tag detection in shared photos, and do the following if a Tag is present:

- In case of a decodable tag, PSPs should follow the policy specified by the privacy tag. If a user's public key is carried in the tag, the protection should be reversible, so that legitimate users, i.e., people holding the corresponding user's private key, can revoke the protection and view the original;
- In case of an undecodable tag or a decodable tag without a public key, PSPs should still protect the privacy but have the freedom in choosing their own ways of protection, *e.g.* non-reversible Gaussian blurring, but a reversible way is recommended.
- Processed tags and their associated faces should be marked (*e.g.* in the image header) to prevent repeated protection from subsequent downstream players. The encrypted obfuscation key should also be contained in the annotation.

Protocol Amendment: Different PSPs may have different technical capabilities in privacy tag detection and decoding. To avoid potential disputation, a third party (e.g., an open-source) implementation should be referenced. Customized implementation should not be worse than that.

In the following section, we will elaborate the technical side of the design, describe actual implementation of all technical modules, and empirical evaluation results to confirm the feasibility of a technical solution.

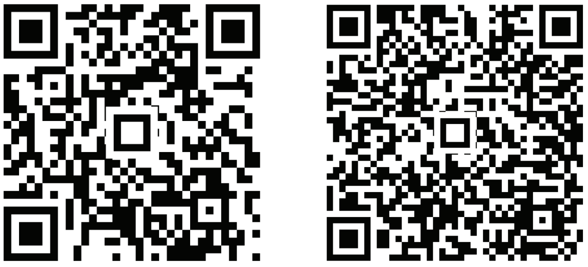

(a) Embedding 32 Bytes (b) Embedding 64 Bytes

Figure 2. Proposed QR-code based Privacy.Tag design, with a color-reversed position locator at the center

4 Privacy.Tag Realization

In this section, we describe and justify our design of a practical Privacy.Tag.

4.1 Required and Desired Tag Properties

Basic Tag Requirements: For a tag to express one's privacy appeal, it needs to fulfill a few basic requirements: First of all, it should be easily detectable and sharply localizable to pinpoint the specific wearer; Secondly, it should work with all cameras, including conventional *e.g.* P&S cameras, DSLRs, and those on mobile or wearable devices. That is, the tag has to activate itself solely via the light medium; Thirdly, it should be able to convey certain amount of information.

Desired Tag Properties: We further desire a Tag to be easy to carry while working in a reasonable range (say a few meters) in real situations, to consume no or little energy, and to be obtainable at low cost. Moreover, a Tag should be as unintrusive as possible, ideally invisible. Reusing exiting familiar tag types will reduce noticeability, but at the risk of confusing with other tags that may appear in physical environments.

4.2 QR-code Based Tag Design

After examining and dismissing many possible candidates such as a mobile phone or RFID, we finally adopt the QR-code as the base of a Privacy.Tag.

Customized Yet Compabitle QR-code: Given the popularity of QR-codes, existing deployed QR-codes on physical objects may trigger false positive detection of Privacy.Tag and lead to undesired protection, *e.g.* taking photos near a poster which contains a QR-code, or while holding a beverage with a QR-code on the container.

To avoid being confused other conventional QR-codes, we customize the design of our QR-code by embedding a special pattern, termed *Privacy.Tag indicator* (PTI), to the center of a QR-code. In particular, we design the PTI to be a *same-sized but color-reversed* position locator, as shown in Figure 2. The PTI has the same size and also the 1:1:3:1:1 proportion-reserving property as normal position locators, and thus enjoys the same detectability as the position locators. We reverse its color (black to white, and vice versa) to avoid confusion with the actual position locators. With this design, we guarantee to reliably tell a Privacy.Tag from

[1] One should avoid using any web sites that may reveal her privacy, which would otherwise lead to even easier privacy leak.

a normal QR-code, as long as the QR-code can be detected, no matter it is successfully decodable or not.

Static and Dynamic Tags: One may print a QR-code based Privacy.Tag and stick it on clothes to express her privacy appeal. This cost is very low. However, as the content is static and unchangeable, it diminishes the control of publicity across different photos. Essentially, when one gives out the private key for one particular photo, she actually gives out control for all photos taken with the same badge. This can be undesirable sometimes, especially when one wants to share only a portion of those pictures.

To pursue fine-grained control of the publicity scope of different photos, dynamic Tags corresponding to different privacy policies and environments can be used. For example, one may design an E-Tag using an E-ink display or even have a smartphone to display a Privacy.Tag when necessary. Latest model of smartphone can show the tag on the screen without incurring too much energy penalty. E-ink is popular for multiple mobile devices for its low power consumption, and we think such E-ink design could be used as a preferable option in producing Privacy.Tag in the future. Obviously, this is a trade-off as E-tags will cost more, while using smartphone will tax energy consumption.

Practical Tag size: It is obvious that the larger the privacy tag, the higher probability it would be detected and decoded. However, people desire the tag to be less noticeable and easier to carry, which favors smaller Tags. According to the state-of-the-art FaceSDK we adopted [12], the minimum size of a detectable face is about 24×24 pixels. Whereas for a QR-code, it requires at least 21×21 pixels to present the whole symbol. Thus, the ideal tag size should be comparable to that of the face to ensure the tag is always detectable whenever a face detected.

However, just as a detected QR-code is not necessarily decodable, a detectable face is not always recognizable. According to face recognition research, the minimum size of a recognizable face is typically about 80×80 pixels, which is also confirmed by our experiments with the auto-tagging feature in Picasa [13]: when the face size is less than 80×80, the uncertainty of autotagging increases dramatically. Considering the fact that the special pattern of QR-codes (i.e., high frequency alternating black and white pattern) makes it easier to detect (not decode), we believe the size of a Tag could be as small as one quarter of the face in area. We will present more detailed study on the impact of tag sizes and justify our decision in Section 7.1.

5 Protocol Realization

In this section, we present an exemplar realization of the key protocol modules to study technical feasibility of the proposed PRSP, as depicted in Figure 3. There are four key functional modules that are unique to PRSP, namely the face/Tag matching, the reversible protection, protection key encryption and the processed Tag annotation. We assume, for photo privacy protection, users want to prevent their face from being recognized.

5.1 Face/Tag Matching

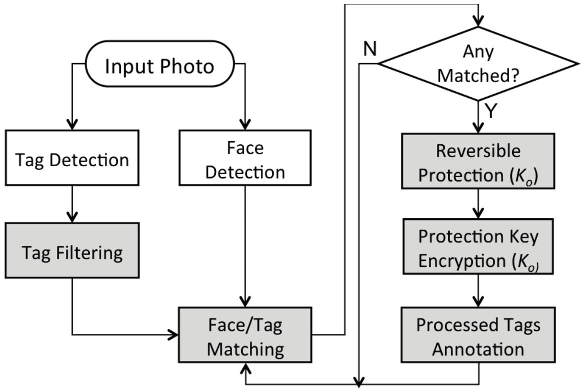

Input Photo
Tag Detection
Face Detection
Tag Filtering
Face/Tag Matching
N
Any Matched?
Y
Reversible Protection (K_o)
Protection Key Encryption (K_o)
Processed Tags Annotation

Figure 3. General privacy protection procedure.

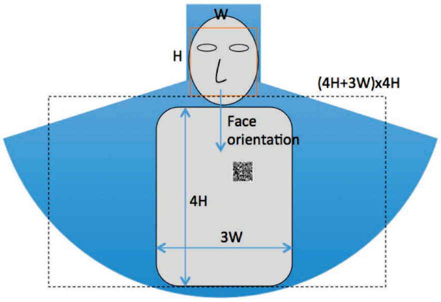


Figure 4. Search area illustration in face/Tag matching. The face and fan-shaped blue areas are the likely tag-appearing area. The face area and dashed rectangular are actually used to simplify the search range.

When a photo is shared, the privacy protector (i.e., PSPs) will perform face detection and also Privacy.Tag detection. More than one faces and tags may be found in the same phone, thus, we need to determine which face a Tag is trying to protect. This is achieved through the face and Tag matching process. Intuitively, if we have effective human body extraction technology, it would be trivial to match a Tag to the face. However, due to various clothes one may wear, body extraction is extremely hard without resorting to depth information or body motion. No mature algorithms can be leveraged, to our knowledge. Therefore, we develop a heuristic algorithm that uses the size and orientation of the detected faces, assuming the Privacy.Tag is worn in the upper body.

Range-constrained Face/Tag Matching: We can obtain from the face detection module the size of a face, say H long and W wide, and the face orientation as determined by positions of the eyes and the nose. Then the user's upper body are around $3W$ wide and $4H$ high. As a normal user can only tilt her head in a limited angle range, say within 60 degrees to left or right, we determine the possible range of a Tag to be a fan-shaped area that spans about 120 degrees, symmetric along the face orientation, with the original at the face center and a radius about $4H$, as shown in Figure 4. In practice, we simply use a rectangular sized $(4H + 3W) \times 4H$ under the face along the face orientation, i.e., the area depicted by the dashed lines in the figure. We also search an extended face area to take care of the case when a Tag is put on a hat.

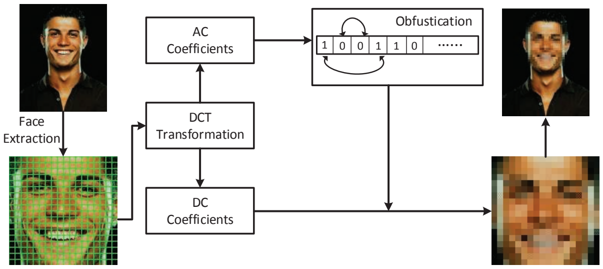


Figure 5. Procedure of proposed secret block-based obfuscation process for privacy protection.

In rare cases, multiple Tags are detected in the effective search region of a face, we empirically select the one that is closest to the face. We also prefer the tag that is directly under the face orientation. This rule is applied when the to-face distances among Tags are similar. We note that, with recent advancement of face recognition, special attributes of faces can be extracted, which can readily tell the gender and even age. Thus, we may include such information in the tag to improve face matching. Richer face recognition features can be stored in the privacy policy site referred by `url`. We leave this for our future work.

5.2 Reversible Protection

Commonly adopted privacy protection strategies include face blurring or mosaicing [6, 28]. Unfortunately, they are lossy processes and irreversible. In PRSP, the protection needs to be reversible to grant the publicity control to the Tag owner. To this end, we protect the privacy through a *secret pattern-guided block-based obfuscation process* that shuffles the frequency components among face area image blocks according to a randomly generated pattern (a binary string termed obfuscation key, K_o, hereafter) by the privacy protector. We elaborate the process using the most prevalent JPEG format.

Block-based obfuscation Process: The proposed obfuscation process is very simple: first sequentially map all bits in K_o to 8×8 image blocks (the basic coding unit in JPEG) in the face area, then exchange *all* the AC coefficients between two image blocks that both map to either a 0-bit or a 1-bit, as depicted in Figure 5. The resulting protected face bears a mosaic looking. The bit stream in K_o is cyclically concatenated in case there are more face image blocks than the length of K_o, (rare).

Our decision of exchanging only, but all, AC coefficients, instead of all DCT coefficients that would be equivalent to shuffling in the spatial domain, is to pursue aesthetic appearance of resulting face-protected images: the face area still looks like a face, but all details are messed up. It also enjoys high operation efficiency, as compared with exchanging only partial AC coefficients, because the zig-zag run-length coding in JPEG makes the coding of AC coefficients interdependent.

We notice that randomly generated K_o tends to have short distance between neighboring 0s and 1s. As the resolution of pictures gets higher, nearby 8×8 blocks will look more similar. Exchanging contents between 8×8 blocks may not be suffice to protect the face. One simple work-around is to group multiple neighboring 8×8 blocks together to form large blocks as exchange units. According to our experiments, such obfuscation process is strong enough to fail most common face detection algorithms, including the FaceSDK we adopt in our work. The bottom right picture in Figure 5 may still be recognizable to human eyes if the subject is expected, even after blurring. This is partly because human recognizes people using additional features such as body shape, clothing, and context, in addition to faces. Based on the current capability of computational face detection by computational methods, we believe that the block-based obfuscation process is a reasonable choice to prevent mass photo labeling.

In addition, a single protection key may be applied for all faces, or different keys may be used for different faces. The former is simpler, but is less strong in privacy protection because any valid private key from any wearer would recover the whole picture and may risk others' privacy.

Reverse obfuscation: Since there is no information loss in the proposed obfuscation process, the protection can be revoked to restore the original face by reversing the obfuscation. It is easy to see that the obfuscation process is symmetric. That is, given K_o, another pass of obfuscation will yield the original photo. Clearly, the strength of protection is controlled by K_o, longer K_o should be used to have stronger protection.

The reversible property of proposed protection strategy actually implies an important benefit to the privacy protectors: it avoids storing the original copies. If an irreversible privacy protection is exerted, then the original copy would have to be retained. Otherwise, criminals would exploit PRSP compatible systems by wearing a Privacy.Tag when committing a crime. Thus, the benefit of reversible protection can be huge for law enforcement purposes.

5.3 Obfuscation Key Encryption

We also hope to give the control of the publicity scope back to the user. This is possible only when we have a way to securely pass the protection key to the user. In our design, we allow a user to specify a public key for this purpose, and design different key protection schemes depending on the decodability of the Tag.

For a decodable Tag containing a user's public key K_{pu}, the privacy protector will use *that* key to encrypt K_o. The resulting encrypted protection key is $K_{eo} = encrypt(K_{pu}, K_o)$. Otherwise, the protector has the freedom to use either a reversible or irreversible protection. As mentioned above, we advocate to still use a reversible protection for storage savings. In this case, the protector will use its own public key K_{pp}, and we have $K_{eo} = encrypt(K_{pp}, K_o)$. As long as a reversible protection is used, the encrypted obfuscation key K_{eo} should be encrypted and embedded into the photo file. Only in this way, a legitimate user can revoke the protection.

5.4 Processed Tag Annotation

A processed Privacy.Tag must be explicitly marked to avoid repeated processing that would lead to wrong and undesired protection when the photo propagates to other

PSPs. In our design, we annotate a processed Tag as follows: $\{(T_x, T_y), [(F_{0x}, F_{0y}), (F_{1x}, F_{1y})], \{\texttt{KeyLen}, K_{eo}\}\}$, where (T_x, T_y) is the center position of the Tag, $[(F_{0x}, F_{0y}), (F_{1x}, F_{1y})]$ is the protected face area obtained from face detection, which is necessary for the revoking the protection, and $\{\texttt{KeyLen}, K_{eo}\}$ are the length and the actual value of encrypted protection key. If there are multiple Tags in a photo, we concatenate their annotations.

Note that, direct editing (*e.g.* resizing, cropping, rotating) or transcoding of protected photos may risk the faces becoming irrecoverable, because it may change the block division and typically involves a re-encoding process. Nonetheless, legitimate users can always edit the original photo. The protection can be exerted again into edited photos, following exactly the same procedure.

6 Implementation

We have implemented the proposed PERP and evaluated various components to demonstrate the feasibility of QR-code based Tag design. We provide some implementation details and evaluation results in this section.

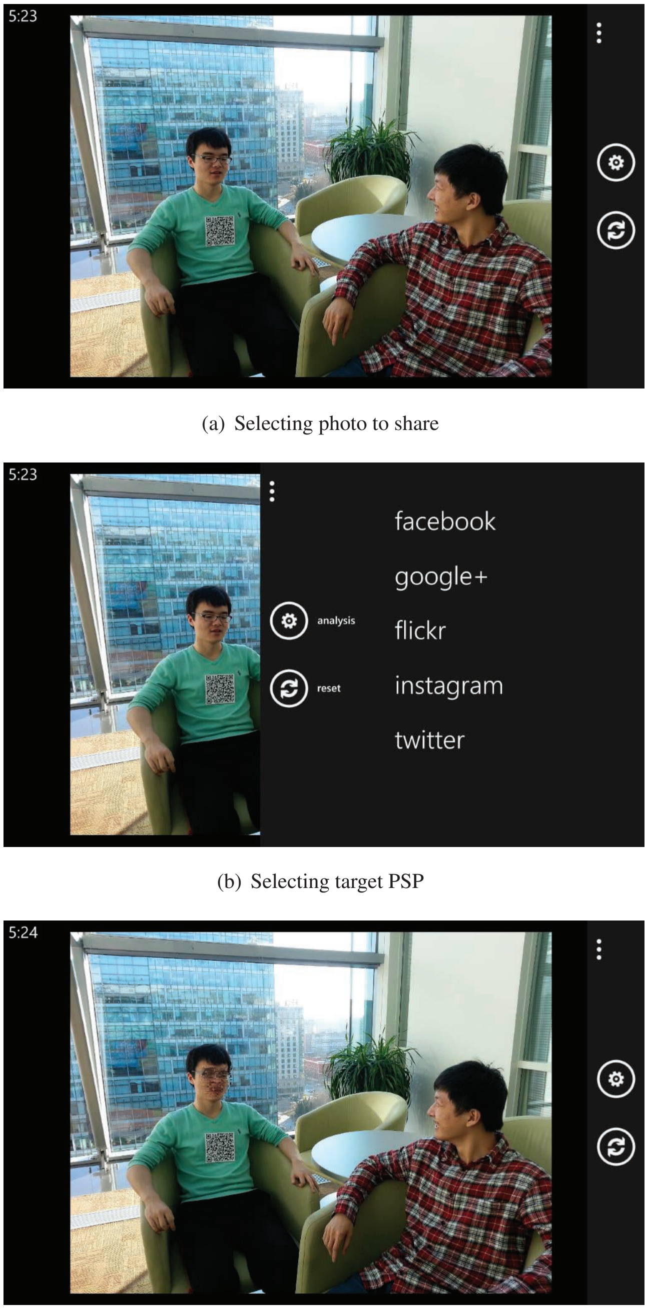
6.1 Key Components

Face Detection: Improve the performance of face detection is out of the scope of this paper. We simply adopted a Microsoft FaceSDK [12], a state-of-the-art face detection tool. This SDK can return the face area via a bounding rectangle and also indicates positions of eyes, the nose and the mouth in each detected face. We have assumed there is no privacy issue if a user's face cannot be detected.

Tag Detection and Decoding: Existing QR readers would fail if it cannot decode a tag even though the tag can be detected. In our case, we need to know if the Tag can be detected no matter whether it is decodable or not. The detection of the Tag is crucial as it can lead to opposite privacy protection behavior. Therefore, we wrote our own Privacy.Tag detector based on the open source implementation ZXing [19]. Our detector not only tells the detectable ones from decodable ones, but also robustly tells a Privacy.Tag from a common QR-code from the color-reversed position locator pattern in the center area of the tag.

Only detected faces might be protected. Thus, in our implementation, we do not detect tags on the whole picture. Rather, we limit it to a small range determined by the faces, as described in Section 5.1.

Tag Annotation Embedding: We leveraged the reserved fields in JPEG picture header to embed the annotations of processed privacy tags. In particular, the JPEG standard allows up to 16 (marked by `X'FFE0'` through `X'FFEF'`) application segments reserved for application use. The special 'Application data syntax' is also defined, consisting of an application data marker, followed by the data segment length and also the application data type. Each such data segments can host up to 64kB data [43]. In our implementation, we have chosen `X'FFEE'` as the marker. In the payload field, we use 16-bits to represent a position element. Assume the encrypted obfuscation key K_{eo} is Len bytes long, then each tag annotation takes $14 + Len$ bytes.



(a) Selecting photo to share

(b) Selecting target PSP

(c) Upon sharing

Figure 6. Privacy.Tag Implementation in Windows Phone 8. The person wears a Privacy.Tag say only Facebook is allowed. When the photo is being shared to Google+, the face is protected

Protection Removal for Legitimate Users: We developed a simple filter that takes in as input a private key and a JPEG file, extracts the Tag annotations from the JPEG header, decrypts the protection key, reverses the obfuscation, and outputs restored original bit stream of the JPEG file. The resulting JPEG file can be viewed normally with any photo viewer.

6.2 Prototype

We have fully implemented the proposed PERP on Windows Phone 8 platform. Figure 6 shows a real use case. The user was about to share a photo of two people in a casual chatting to Google+. One of the two was wearing a Privacy.Tag, saying only allow to show unprotected face on Facebook. Thus, upon sharing, the person's face are protected.

7 Evaluation

Our evaluation focuses on characterizing various properties of the QR-code based Privacy.Tag design and also the computational overhead of face and Tag detection and the actual face protection process. We use Samsung Galaxy S3 that features a lens with 40mm focal length and 8M pixel resolution in all our experiments.

We emphasize that all the distance related experimental results should be referred to w.r.t this 40mm focal length as different camera focal lengths have different magnification factor and affect the working ranges. An object will appear larger with a telephoto lens than a wide angle one at the same shooting distance. Nonetheless, the relative size among objects (e.g., faces and tags) remain the same. Conclusions derived on relative size will still hold. In addition, most phone cameras have focal lengths close to 40mm, as its field of view is close to that of human visions.

7.1 Tag Effectiveness

In this subsection, we mainly evaluate the performance of the Tag detection algorithm under different shooting conditions. As is known that QR-code detecting and decoding is no longer a novel technique, our evaluation is mainly to provide a sense that given the current state-of-the-art QR-code detection tools, how large a QR-code should be in order to reliably detect them to provide effective privacy protection.

Effective Range vs Tag Size: Different sized Tags will have different working ranges. We want to find a proper size that is suffice for face protection purpose. To this end, we first measure the scale of both a face and different sized Tags in real photos taken at various distances. We asked one user to wear Tags by sticking his T-shirt with side length 5cm, 10cm, 15cm, and 20cm, and took photo across distances from 1m to 15m at step of 1m. The venue is a hallway with a mixture of indoor and outdoor lighting.

Figure 7 shows the results, where the sizes (in pixels) are the side length of the bounding boxes returned by the FaceSDK and our QR detector. We only plot up to the range that a face or Tag can be detected. We see from the figure that people's faces can be detected in up to 12m, where the face image size is around 38 pixels. The minimum image size a Tag is detectable is about 30 pixels. This means, as expected, larger Tags will have larger working ranges, e.g., a 5cm Tag can be detected at 5m whereas a 20cm Tag are still decodable at 11m.

The minimum size for a face to be recognized is about 80 pixels by the FaceSDK and also confirmed with Picasa's autotagging feature.[2] This corresponds to about 5m shooting distance, as indicated by the horizontal dashed line. At this distance, all Tags can be detected, hence, are able to provide privacy protection.

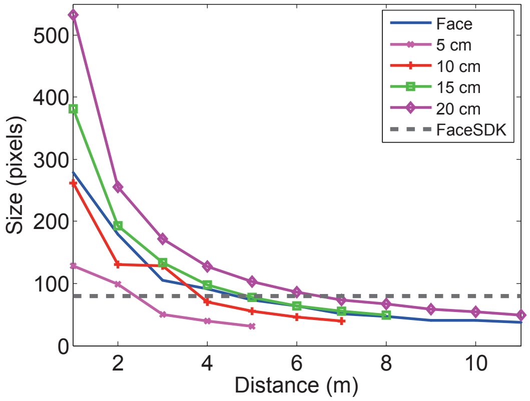


Figure 7. Size of face and tags in the photos taken at different distances.

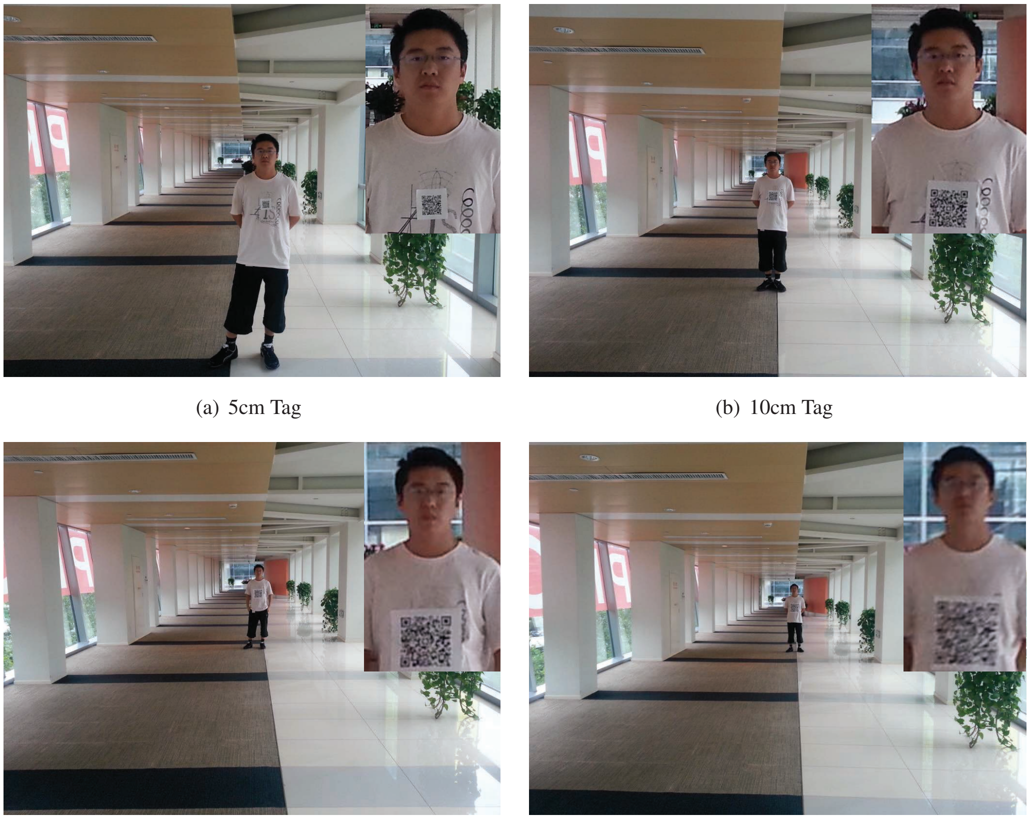
We notice that the detectable image size for both faces (38 pixels) and Tags (30 pixels) are slightly larger than what the FaceSDK claimed (24 pixels) and ideal QR-code (21 pixels), respectively. The reason might be that the camera shake as we hand held the phone, and the imperfect auto-focusing of the phone camera. But we believe they represent the actual performance of FaceSDK and our QR detector in real situations.

To illustrate the real situations, Figure 8 shows pictures with different Tags at their maximum detectable distances. We also show the portion of the face and upper body cropped out from the picture displayed at the actual size in the top-right corner. Evidently, the face becomes more blurred when the shooting distance increases.

Tag Detectability and Decodabilility: Previous experiments show the maximum detectable ranges for different Tag sizes, in which one successful detection out of 10 trials is considered detectable. Now, we further examine the actual detectability and also the decodability at different distances. Figure 9 shows the detection probability for different Tag sizes (all carrying 32 Bytes information, as shown in Figure 2(a)) across different distances in both indoor and outdoor environments. The vertical lines indicate the maximum decodable distances below which Tags cannot be reliably decoded. As expected, the larger the Tag is, the more likely it is detectable and decodable.

From the figures, we can see that the longest detectable distances for 5cm Tag are 4m for indoor and 6m for outdoor scenarios, and those for the 10cm Tag are 7m and 8m, respectively. The 5cm Tag can be reliably decoded at a shooting distance of 2m for both indoor and outdoor, and the detecting rate is about 80% at a distance of 3m (indoor) and 90% at 4m (outdoor). The reliable, decodable distance for 10cm Tags will increase to 3m for both indoor and outdoor

[2] A face smaller than 80 pixels may still be recognized by human eyes. However, it is difficult to get a consensus through user study that the blurred face is strong enough to prevent the user from being recognized. An important fact is that human recognize people not only from protected face. Therefore, in this paper, we rely on objective technical measurement and avoid subjective evaluation by humans.



(a) 5cm Tag

(b) 10cm Tag

(c) 15cm Tag

(d) 20cm Tag

Figure 8. Sample pictures showing different sized Tags at their maximum detectable distances.

settings, and the detecting rate is about 90% at 5m (indoor) and 75% at 6m (outdoor). The reliable, decodable range dictates the *physical range* in which when a photo is taken, the user can control the publicity scope of the resulting photos via the Privacy.Tag.

Information Embedding Capability vs Distance: The amount of information embedded in a QR-code is determined by the density/complexity of QR-code, which maps to different versions of QR-codes. The less information, the simpler and lower version of the code. Obviously, given a fixed size, simpler codes will enjoy larger detectable and decodable distances. Our design of Privacy.Tag embeds a special pattern at the center of a QR-code. The patterns are intentional errors. While they are correctable, they consume additional protection bits. This leads to increased complexity (or version) of QR-codes, and is the cost we pay for better disambiguation from other QR-codes.

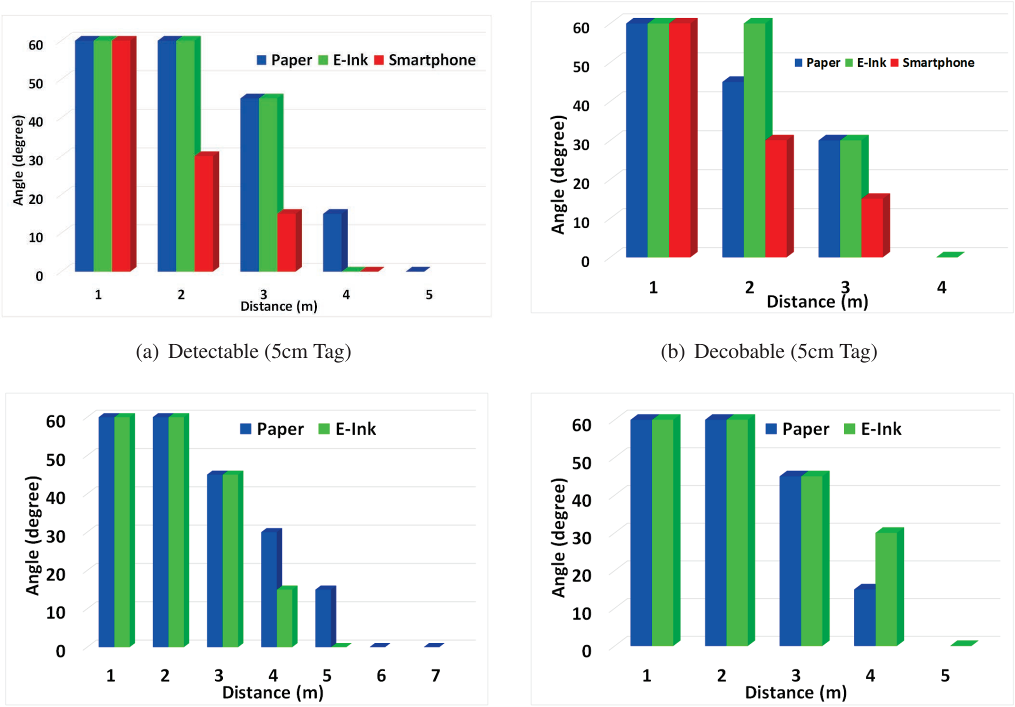
In this experiment, we evaluate the capability of decoding Privacy.Tag with different amount of content, namely 16, 32 , 64, and 128 Bytes, and test the real decodable distances in different environments. To put into perspective, 32 Bytes can carry a shortened URL and two popular domain names. We also test the performance when the Tag is presented via E-ink display (Kindle) and smartphone (Lenovo S920). For 10cm Tags, we only test on the paper and E-ink as the smartphone screen is not enough large.

Table 1. Reliable Decoding Distances vs Amount of Embedded Information, for 5cm and 10cm Tags.

	Materials	16B	32B	64B	128B
5cm	Paper Tag	3m	2m	1m	1m
	E-Ink	4m	3m	2m	1m
	Smartphone	3m	2m	1m	1m
10cm	Paper Tag	4m	3m	2m	2m
	E-Ink	5m	4m	3m	2m

Table 1 shows the reliable, decodable distance for different QR-codes densities, carried by the three media. We can see that the more information one embeds, the smaller, the reliable, decodable ranges. Note that by increasing in QR-code density, the position locator will become smaller, hence the detectable ranges will also be affected, but the extent is much lighter thanks to its strong error correcting pattern.

Impact of Shooting Angles: Shooting angle affects Tag detectability and decodability as well. We test the detectability using the 5cm Tag (still carrying 32 Bytes information) at different shooting angles, ranging from 0 degree to 60 degrees (which are normal range one turns his head) at steps of 15 degrees, at different shooting distances, and for the three media for a 5cm Tag and two media for a 10cm Tag. Results are shown in Figure 10. All three kinds of Tags could be reliably decoded when the shooting distance is one me-



(a) Detectable (5cm Tag)

(b) Decobable (5cm Tag)

(c) Detectable (10cm Tag)

(d) Decobable (10cm Tag)

Figure 10. Tag detecting and decoding at different angles across different shooting distances.

ter, even when the angle is about 60 degrees. The decodable angle shrinks when the shooting distance increases. For the 5cm Tag, at 3m distance, the Tag can be detected (but not reliably) and decoded at an angle up to 45 and 30 degrees, respectively. For the 10cm Tag, at similar angles, the range extends to 4m. The overall performance of E-ink is similar to paper Tag, while the smartphone screens are not as good. The reason is that smartphone screens are more reflective than E-ink screens. Somewhat surprisingly, E-ink displays slightly outperform paper in some cases, thanks to its always-flat screen whereas the paper Tag stuck to T-shirt may be crumpled.

7.2 Face Protection

One key challenge in our proposed PRSP is to reliably match a Tag to the right face, especially when dealing with a group of people, some of whom wear Tags and the rests don't. We have proposed a range constrained fact/Tag matching heuristic. We evaluate its performance with two sets of experiments to cover both indoor (Office) and outdoor (Park) environments.

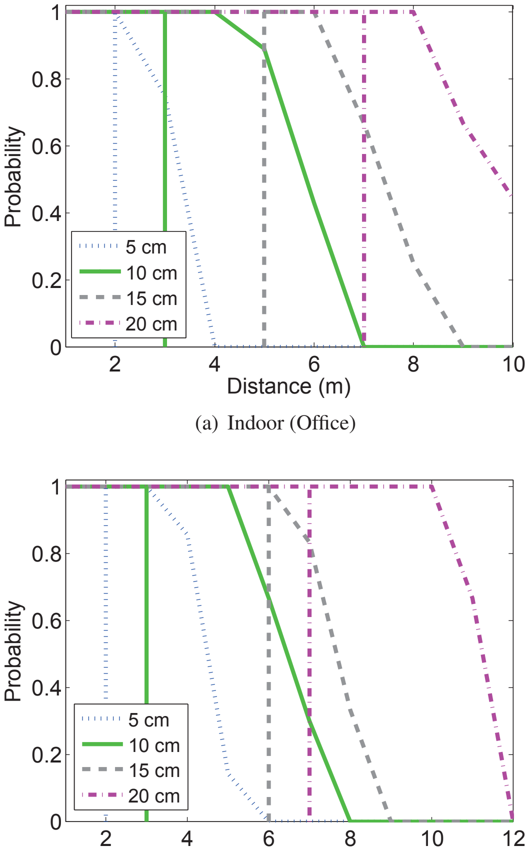
Experiment Settings: We gathered 5 people to participate a small group discussion in a meeting room and then asked them to a group tour in the garden. Among them, three people wore 5cm Privacy.Tags and the rest two did not. We did not convey the purpose of the experiments, and asked them to behave as usual. We took photos from different angles freely. As we checked the resulting photo set (about 120 photos), they actually cover many challenging cases such as one's tag was blocked by other people, and someone was side facing the camera, among others.

Performance Metric: We consider Tag wearer' faces being protected by their own Tags (i.e., a Tag matched to the right face) as true positive, and non-Tag wearers' faces remaining public (i.e., not protected) as true negative, whereas Tag wearers' faces not being protected as false negative, including both cases of face/Tag mismatching and failure Tag detection for a detected face, and non-Tag wearers' faces being protected as false positive. Then we define two performance metrics: *precision* equals to the number of true positive cases divided by the sum of true positive cases and false positive cases; and *recall* equals to the number of true positive cases divided by the sum of true positive and true negative cases.

Table 2. Face/Tag Matching in Real Situations

	Precision	Recall
Indoor	96.20%	77.22%
Outdoor	77.42%	78.26%

Results: The precision and recall for both indoor and outdoor scenarios are presented in Table 2. We find that the precision is surprisingly high (96.2%) and the recall is relative low for indoor cases, whereas both metrics are relatively low for outdoor cases. We confirmed that, despite the different lighting conditions, they are bright enough and are not the main factors affecting the performance. The high precision and low recall in indoor environment is mainly due to the limited room size that, on the one hand, limits the shooting distance to be small, and on the other hand, causes more oc-



(a) Indoor (Office)

(b) Outdoor (Park)

Figure 9. Detection probability different sized Tags (carrying 32Bytes) in indoor and outdoor environments.

clusions due to causal poses and gestures, and larger shooting angles. The major influence factors for the outdoor cases come to people's motion in the pictures, non-steadiness (due to walking) when taking pictures, and partial face blocking.

A necessary condition for privacy protection in group photos is to match the Tag to the right wearer. Therefore, we separate the concerns between face association and tag decodeability in these two cases.

7.3 Computational Overhead

We advocate PSPs to support proposed PRSP. Having argued that they do not need to save additional copy of photos (thanks to reversible protection scheme), the only concern is the computational overhead. We thus measure the computational overhead, using a Desktop PC with i7-2600 CPU and 8GB memory, running Windows 8.

Table 3 shows the breakdown of the time consumed by key PRSP modules. We see from the table that the face detection takes approximately 2.393 seconds for 8M pixel photo size. Note that face detection time is mostly determined by the image size, and not affected by the number of faces in the photo [12]. The detection of Privacy.Tags only takes 0.01 seconds, thanks to the range-constrained detection mechanism, and the decoding consumes 0.029 seconds. We do not measure the time for privacy policy retrieval as it is highly affected by network conditions. As the policy can be cached and indexed by Tag contents, the time should be similar to that of DNS resolution, which is usually few hundred milliseconds [38]. The obfuscation process costs 0.068 seconds. We also show the time for the protection restoration, which is simply another block shuffling process. The time is 0.07 seconds, similar to that of protection as it is a reversible, symmetric process. Mainstream PSPs are capable of conducting face detection and have already put them in production systems (e.g., auto tagging), the increment on computational cost for implementing the proposed PRSP is thus negligible.

Table 3. Computation Time Breakdown of Major Modules of the proposed PRSP.

Module	Time (seconds)
Face Detection	2.393
Privacy.Tag Detection	0.01
Privacy.Tag Decoding	0.029
Face Protection	0.068
Revoke Protection	0.07

Considering the fact that people usually take pictures in short distances, *e.g.* a few meters, our experimental results lead to the following conclusions: 1) it is necessary to protect the face even when Tags are not decodable; and 2) a 5cm or 10cm Tag is a practical choice for Privacy.Tag for their ability to protect the privacy and gain effective control of the publicity scope, in addition to its convenience of carrying; 3) our QR-based Tags are equally effective on different display media (paper, E-ink or phone screen); and 4) our proposed PRSP protocol incurs negligible overhead for PSPs that already deploy face-detection based features.

8 Discussions and Future Work

We have presented our exemplar design of QR-code based Privacy.Tag design and evaluated various aspects of it. In our current implementation, the recall rate and the precision (especially for outdoor scenarios) in our evaluation are not very high due to various challenges, such as occlusion and blurry picture. Generally, blockages and occlusions may occur frequently in real world application, and yet the faces are still visible in the pictures to human user. One possible solution is to place a tag directly on a person's face, which makes them quite intrusive. An investigation into invisible tags - tags that are invisible to human eyes yet can be detected in camera images - is of our interests.

According to our evaluation in real scenarios, sometimes the Tags are partially blocked or the photo is blurry. In this case, more robust tag detection algorithms can be studied to detect and decode fragmentary or blurry QR-code in the photo. Hence, if a user has a very strong appeal of privacy, he/she should wear the Privacy.Tag on more obvious location on the body to avoid blockage, or wear several such Tags. In other words, simply wearing a tag does not guarantee protection. Our proposal provides a way for users to express their privacy desires, and only if such desires are expressed properly will PSPs be able to respect them. In this case, the evaluation results still confirm the feasibility of the current solution. We do not attempt to claim that it is the only viable solution - there still exist plenty of scope for improvement. Similarly, we have designed a reversible protection scheme through a secret obfuscation process. Many alternative or better ways can be designed.

We have adopted the standard QR-code, which has a very limited capacity, which has in return severely constrained the amount of information we can put in tags and the decodable ranges. Our design of incorporating a special Privacy.Tag indicator, which consists of intentional errors, further exaggerates the issue. If higher capacity codes are used, or dedicated Tags are designed, the problem would be simpler. We could beautify the current QR code by introducing stylish design [25] without affecting the performance, such as Halftone QR Codes [27], Visualead QR codes [17]. Or we could also adopt new kind of picture-embedding 2D barcode, such as PiCode [31].

We assume anonymous privacy policy hosting services. However, if the PERP is widely adopted, those services can become a bottleneck, given the huge number of photos taken daily. Scalable network architectures similar to the DNS service may need to be imposed.

Another possible concern between a user and the PSPs would be that the user may insist that the Tag in the picture is obvious enough yet the PSP does not detect the Tag and respect the privacy desire properly. This may become more of an issue if the PSPs offer the privacy resection and protection as a charged service. However, a possible solution would be to establish a third-party equipped with reliable state-of-the-art Tag detection system as an arbiter, and the PSPs should implement a Tag detection system with the performance no worse than the arbiter's.

The propose scheme will work for normal benign photographers, but not for professionals like the paparazzi who may avoid capturing the Privacy.Tag while taking photos or simply remove the Tag from the photo before sharing. The security level of protection depends on the rule of obfuscation process and the length of the random pattern.

While we have adopted the QR-code as a concrete embodiment of a Privacy.Tag, we are not limited to QR-codes. Although not everyone is willing to wear QR-codes all around their outwears, we still think that people may adapt their behavior when there are new appearances, provided the new approaches can bring value to them. For example, bring significant value to those who have privacy concerns, which has become a top concern nowadays. We also notice that the QR-code is indeed universal and people have started to have such codes on their clothing either for fun or for advertisement purposes. For future work, designing a higher capacity, more stylish or less noticeable even invisible Privacy.Tag is of great interests. Clothes or accessories are designed nowadays that integrate sophisticated techniques and ideas, such as the Fibonacci scarf [3]. This kind of design may offer new opportunities. We expect other forms to emerge, provided that the proposed concept is accepted. Maybe in the near future, more sophisticated or invisible tags are designed, which not only protect people's privacy according to their individual desires, but also new fashion trends. Robustly matching the Privacy.Tag to the right face is a fundamental challenge that deserves more investigation. As aforementioned, revising the content of the tag, to embed some face attributes (*e.g.* eigenface) into the Tag or on the privacy profile site, can be an effective solution.

9 Related Works

Privacy protection is a broad topic and have attracted extensive research attentions. There are much research work on how the privacy is revealed when sharing photos online [20–23, 30, 32, 44]. Our work is orthogonal to those efforts. Our focus is on how to let the users explicitly express their diverse privacy appeals and how PSPs should react to respect them. To our knowledge, this is the first effort along that new direction.

There are also efforts on how to protect the privacy by concealing persons, blurring, masking, mosaicing the selected areas (mostly faces) of images [6, 28, 35]. Our reversible face protection is also quite different from those lossy methods that cannot restore the original, rooted from the goal to give the publicity control of photos back to the user. A piece of related work is P3 [36] that extracts and encrypts small components of photo while preserving the rest in public. P3 works well only if the photo owner/shares happen to be the user self. Our work aims at a systematic privacy protection solution, and gives privacy control to the user no matter who takes and shares. Face blurring has been used as one form of denaturing in GigaSight [42]. They propose that denaturing may not only involve content modification but may also involve meta-data modification because of the fact that people's privacy may still be compromised when videos are taken at the same placesor time from other users with different privacy settings. In this case, this work either blanks the frame completely or passes the frame through unmodified and any faces being detected will be blurred, which could be considered as a rigorous treatment of online privacy.

Some new tags for the privacy protection purpose are emerging. PriSurv [26] utilizes RFID techniques to control the personal information disclose. Glasses equipped with near-infrared LEDs that emits invisible light but can be captured by camera is designed to convey hidden privacy appeal of not taking photos of me [45]. These tags need to work with either instrumented surveillance systems or with smart cameras, whereas our design of Tag makes no assumption on cameras. Nonetheless, these work can be incorporated into our system. TagMeNot also uses QR-tags to let people express their privacy concern and calls for photo-takers to avoid taking photo of them [16]. Their proposal shifts the burden to the photo-taker. Our design is about systematic and automatic privacy protection solution involving mainly PSPs, without human in the loop.

10 Conclusion

In this paper, we have presented the Privacy Expressing and Respecting Protocol that represents a new privacy protection paradigm that gives privacy control back to the users. It consists of two components, the Privacy.Tag and the associated Privacy Respecting Sharing Protocol. The Privacy.Tag is a wearable tag that enables a user to explicitly signal her privacy appeal and to express her own privacy policy via simple syntaxes. The PRSP is a set of simple rules that regulates photo service providers (PSPs) to respect user's privacy policy specified in the Tag. It protects Tag wearer's privacy by default, and protects the face area with a reversible obfuscation process. The obfuscation key is encrypted with user's public key contained in her privacy policy. With this design, the user can restore the original photo, and can control the publicity scope of the photo by controlling the dissemination of her private key. We have fully implemented the PERP, and evaluated various aspects of our Tag design, the protection performance and the computational overhead of the Protocol. Our results confirm the technical feasibility of PERP. We advocate PSPs to collectively follow the Protocol and contribute to a healthy photo sharing ecosystem.

11 Acknowledgments

The work of Bo is supported in part by the US National Science Foundation under Grant No. CNS-1319915 and CNS-1343355. The research of Li is partially supported by NSF CNS-1035894, NSF ECCS-1247944, NSF CMMI 1436786, National Natural Science Foundation of China under Grant No. 61228202.

12 References

[1] Camera phone predator alert act. http://www.govtrack.us/congress/bills/111/hr414#citations/.

[2] Congressletter. http://blogs.wsj.com/digits/2013/05/16/congress-asks-google-about-glass-privacy/.

[3] Fibonacci scarf. https://www.dianaeng.com/smart-scarves/.

[4] Google asked me to remove glass at their own event. http://www.geek.com/android/the-very-first-time-i-was-asked-to-remove-my-glass-was-by-google-1564076/.

[5] Google glass. http://www.google.com/glass/start/.

[6] Google street view. http://www.google.com/streetview.

[7] Google street view privacy and security. http://www.google.com/maps/about/behind-the-scenes/streetview/privacy/.

[8] How many photos are uploaded to flickr every day, month, year? http://www.flickr.com/photos/franckmichel/6855169886/.

[9] Instagram press center. http://instagram.com/press/.

[10] Japanese iphone makes loud shutter sound in silent mode. http://www.quickonlinetips.com/archives/2008/07/japanese-iphone-makes-loud-shutter-sound-in-silent-mode/.

[11] Memoto. http://memoto.com/.

[12] Microsoft research face sdk. http://research.microsoft.com/en-us/projects/facesdk/.

[13] Picasa. http://picasa.google.com/.

[14] Privacy international. https://www.privacy-international.org/reports/south-korea/ii-surveillance-policy/.

[15] Stop the cyborgs. http://stopthecyborgs.org/.

[16] Tagmenot.info. http://tagmenot.info/.

[17] Visualead. http://www.visualead.com/.

[18] The web robots pages. http://www.robotstxt.org/.

[19] Zxing. http://code.google.com/p/zxing/.

[20] A. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *PET*, 2006.

[21] S. Ahern, D. Eckles, N. Good, S. King, M. Naaman, and R. Nair. Over-exposed?: privacy patterns and considerations in online and mobile photo sharing. In *CHI*, 2007.

[22] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. In *SIGCOMM*, volume 39, pages 135–146. ACM, 2009.

[23] A. Besmer and H. Richter Lipford. Moving beyond untagging: photo privacy in a tagged world. In *SIGCHI*, pages 1563–1572. ACM, 2010.

[24] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07*.

[25] H. Blasinski, O. Bulan, and G. Sharma. Per-colorant-channel color barcodes for mobile applications: An interference cancellation framework. *Image Processing, IEEE Transactions on*, 2013.

[26] K. Chinomi, N. Nitta, Y. Ito, and N. Babaguchi. Prisurv: Privacy protected video surveillance system using adaptive visual abstraction. In *Advances in Multimedia Modeling*, 2008.

[27] H.-K. Chu, C.-S. Chang, R.-R. Lee, and N. J. Mitra. Halftone qr codes. *ACM Transactions on Graphics (Siggraph Asia)*, 2013.

[28] F. Dufaux and T. Ebrahimi. A framework for the validation of privacy protection solutions in video surveillance. In *ICME*, 2010.

[29] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS*, pages 89–98. ACM, 2006.

[30] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *WPES*, pages 71–80. ACM, 2005.

[31] W. Huang and W. H. Mow. Picode: 2d barcode with embedded picture and vicode: 3d barcode with embedded video. In *MobiCom*, pages 139–142. ACM, 2013.

[32] J. M. Kleinberg. Challenges in mining social network data: processes, privacy, and paradoxes. In *SIGKDD*, pages 4–5. ACM, 2007.

[33] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing facebook privacy settings: User expectations vs. reality. In *ACM IMC*, 2011.

[34] J. Palank. Face it:bookno secret to employers. *The Washington Times*, 17, 2006.

[35] A. Poller, M. Steinebach, and H. Liu. Robust image obfuscation for privacy protection in web 2.0 applications. In *IS&T/SPIE Electronic Imaging*, 2012.

[36] M.-R. Ra, R. Govindan, and A. Ortega. P3: toward privacy-preserving photo sharing. Technical report, 2013.

[37] P. Raikonen. Robots exclusion protocol. *Communications of the ACM (in printing)*, 2009.

[38] V. Ramasubramanian and E. G. Sirer. Beehive: O (1) lookup performance for power-law query distributions in peer-to-peer overlays. In *NSDI*, volume 4, pages 8–8, 2004.

[39] A. Reznichenko, S. Guha, and P. Francis. Auctions in do-not-track compliant internet advertising. In *CCS*, pages 667–676. ACM, 2011.

[40] A. Romano. Walking a new beat: Surfing myspace. com helps cops crack the case. *Newsweek, April*, 24:48, 2006.

[41] J. Rosen. The right to be forgotten. *Stanford Law Review Online*, 64:88, 2012.

[42] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan. Scalable crowd-sourcing of video from mobile devices. In *MobiSys*, pages 139–152. ACM, 2013.

[43] I. Telegraph, T. C. Committee, et al. Information technology-digital compression and coding of continuous-tone still images-requirements and guidelines. *International Telecommunication Union*, 1992.

[44] K. N. Truong, S. N. Patel, J. W. Summet, and G. D. Abowd. Preventing camera recording by designing a capture-resistant environment. In *UbiComp 2005: Ubiquitous Computing*, pages 73–86. Springer, 2005.

[45] T. Yamada, S. Gohshi, and I. Echizen. Use of invisible noise signals to prevent privacy invasion through face recognition from camera images. In *MM*, pages 1315–1316. ACM, 2012.