PRIVACY PRESERVATION IN MOBILE COMPUTING AND NETWORKING: ACCESSING, SHARING AND BROADCASTING

by

Cheng Bo

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2016

Approved by:

_____
Dr. Yu Wang

_____
Dr. Weichao Wang

_____
Dr. Aidong Lu

_____
Dr. Jiang Xie

_____
Dr. Xiang-Yang Li

ABSTRACT

CHENG BO. Privacy preservation in mobile computing and networking: accessing, sharing and broadcasting. (Under the direction of DR. YU WANG)

The ever increasing proliferation of mobile devices has led to unprecedented concern on privacy infringement, and various security threat and privacy leakage emerged from the water in the last few years. Although the rich functionality has allowed users to store personal information and interact with privacy sensitive applications on the devices, it makes them the hot target inevitably for adversaries. Aside from providing local application and storage, the online photo and video sharing and displaying service have also triggered an outcry of concerns about privacy and copyright from the public, especially with new features of automatic tagging and facial recognition. Modern mobile devices usually lock themselves to prevent unauthorized users to access the privacy information stored locally, online social networks use techniques to conceal people's privacy online, such as blurring, masking, and denaturing, or adding hidden watermark into the media to claim the ownership of the digital properties. In this research, I will mainly focus on analyzing the privacy issue in both local and online activities, propose frameworks for preventing information being compromised by attackers and unauthorized users, protocol and prototype for creating a healthy online photo and video sharing ecosystem in long run.

ACKNOWLEDGMENTS

This dissertation could not have been written without the wonderful instructions, consistent encouragement and support from my advisor Professor Yu Wang and Professor Xiang-Yang Li from Illinois Institute of Technology. I am also grateful to Dr. Guobin Shen and Dr. Jie Liu for their insights and invaluable guidance during my work at Microsoft Research Asia. In the past five years, the members of research groups from both UNCC and IIT, and Dr. Lan Zhang from Tshinghua University have helped me succeed through my PhD study, I appreciate all the valuable advices and partnership. Particularly, I am thankful to other members of my dissertation committee: Professor Weichao Wang, Professor Aidong Lu and Professor Jiang Xie, for their services, and help.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The rapid development in smart devices has been stimulating the blooming of the personalized applications and online activities [33], including checking emails, enjoying personal photos and videos, online sharing, mobile payment *etc.*. As the smart devices getting involved into people's daily lives, the device owners face increasing risk of privacy leakage, especially with the data related users' personal information (*e.g.*, text, email, historical records, pictures, and videos). However, with the blooming data oriented context-aware mobile applications [51], the storage of personal information is no longer constrained locally, but in public, such as sharing and broadcasting in online social networks, Photo Service Providers (PSPs), *etc.*. Although different mechanisms are developed to protect local personal data, such as fingerprint, drawing pattern on screen or facial recognition, a large number of users still concern about their data privacy and integrity, especially when the device is shared to other guest users [61, 62, 64]. On the other hand, the ease of taking and sharing photos and videos, along with new features of automatic tagging and linking to one's online profile, have triggered an outcry of concerns about privacy form the public [32]. Under this circumstance, the personal data is difficult to be protected once it is shared or broadcasted into current control-less online ecosystem or in public, so that the content of published data is in huge risk.

In order to alleviate the potential privacy threat during the personal data propagation, and grant users the capability of privacy control without extra cost, designing and implementing

a full scope of protocol to protect people's privacy and ownership is a must.

1.1 Existing Privacy Issues and Motivation

Modern smart devices are becoming increasingly powerful so that programs used to run on desktops are available widely in smart mobile devices nowadays. Currently, the migration allows people to store large amount of sensitive information (such as texts, emails, pictures *etc.*) in the device, or access to online personal account, share photos and video to the online social networks, or present multi-media data in the public. However, the personal information also has great potential threat of being compromised since the moment the data is generated. And with the data propagates from local device to the public, the probability of privacy leakage grows exponentially. In this proposal, I will describe the existing privacy issue in each stages, and address my motivation to handle different privacy issue in each stages so as to provide a full scope privacy preservation protocol.

(1) User identification and authentication for smart devices:

Since initially the data is produced in the devices, safeguarding the private data stored on such smart mobile devices from unauthorized user therefore becomes crucial. User identification has been working as an important component of smart devices for personalized services and secure data access. PIN codes, password or drawing patterns are the most common identification and access control strategies in almost every commercial smart devices operating systems. However, such unlocking schemes have two main drawbacks. First, these unlocking actions is vulnerable for shoulder surfing attack, which happens quite often in public either purposely or unintentionally [73,94,96,100]. Second, the touch screen based devices are susceptible to smudge attack where imposters hack the unlock key through the smudge left on the touch screen by recent user's input [25,27,114]. Meanwhile,

frequently entering passcode and drawing pattern on the screen are always labor-intensive and time-consuming so that some users may leave their device vulnerable. Although more intelligent strategies has been proposed to identify legitimate users, such as facial recognition [37], the accuracy unreliable with changing environment in real application, and it is still annoying and power consuming to take picture frequently. Latest techniques exploit the capacitive touch communication to distinguish different users while they are interacting [104], which extended touch-screen device as a receiver for identification code transmitted by a identification hardware token. Such mechanisms require extra hardware which diminish the conveniency and availability. However, most of these mechanisms have potential risk of being imitated, *e.g.*, peeking over the shoulder, using a photo to cheat the camera, or eavesdropping the communication between devices and token.

(2) Privacy preserving photo sharing:

With the proliferation of smart mobile devices with onboard camera, high-bandwidth mobile network, and online social network and PSPs, taking and sharing photos and videos have become easier than ever. For example, the emerge of wearable devices such as Google Glass [4] and Memoto [10] have accelerated the trend since the photo could be taken quietly and spread online automatically without human in the loop. The latest survey indicates that approximately 1.4 million photos are uploaded to Flicker every day [7], and 40 millions to Instagram [8]. However, the ease of photo taking and sharing activities has triggered an outcry of privacy concern from public because of the new feature of online social networks for facial recognition, automatic tagging and linking to one's online profile. Currently, the photo and video privacy control solutions put a cognitive burden on the subject being photographed [32], and most countries have enacted laws and regulations to enforce noticeable

cues to show the capturing or recording are in action, so that give the subject an opportunity to adjust their behaviors accordingly. Nevertheless, it is still difficult for people to keep track of which device nearby maybe recording or where the photo will end up at, and it is also struggling for most people to configure their online photo sharing policies correctly [71]. Under this circumstance, my question is: do people have to relinquish their privacy in the era of mobile and wearable computing?

(3) Privacy preserving public video broadcasting:

The rapid spread of camera-enabled mobile devices or wearable devices also augments the conveniences of video recording and re-displaying. On the other hand, the blooming of electronic visual display system deployed for various purpose accelerates the information exchange between the visual display and the audiences. Researchers recently propose to encode information into original video by taking advantage of the extra signal which could only be captured by off-the-shelf cameras but not human eyes. A number of innovative systems have been deployed to implement and improve the visual communication over screen-camera links [52, 55, 56, 70, 82, 105, 106, 111]. However, none of these works consider the privacy concern regarding unauthorized pirate video taping behavior, and such privacy concern get severer when personal video is shared online or broadcasted in the public. Therefore, a realistic problem arise: how to prevent unauthorized user from video taping a personal video displayed and broadcasted in public for high-quality redisplay while do not affect the viewing experience for live audiences. Borrowing the idea of filing a copyright for a digital property from film industry, we claim that the video is protected by law and unauthorized usage is illegal. Watermarking is usually added to the original digital property to claim the ownership of the released digital property when broadcasting [39, 109, 110].

Alternatively, inserting extra frames to obscure a recording [115] or projecting ultraviolet or infrared light onto the screen could also wash out recorded pictures [13]. Although such pirate videos contain both valuable video and large amount of obscure images or shades, the content of the video could still be more likely received by human eyes in large extent, needless to say some of the techniques cannot be adopted in other display systems, such as large LCD monitors for personal usage. Under this circumstance, the people's privacy could also be compromised if the video is not dealt properly when displayed in public or broadcasted. Therefore, I propose a universal technology which can be used to protect the video displayed in various devices without introducing extra hardware from pirate video-taping using typical mobile devices, such as smartphones or smart glasses.

1.2 Main Challenges

Many challenges have to conquered so as to achieve the proposed protocol.

(1) Implicit user identification via behavioral biometrics:

Our preliminary investigation indicates that the people's using habits are difficult to be copied, which could be considered as one possible solution for user identification. In order to characterize people's unconscious using habits accurately, we take both user's action and device's reaction into account, including the coordinate of the touch point on the screen, the duration of the touch, the strength, and the vibration and rotation of the device generated by the interaction, *etc.*. In addition, the connection between features may not be neglected. For example, different interaction coordinates on the touch screen may lead to various amplitude of vibration of the devices. The legitimate user's interacting behavior model is easy to establish, since there are abundant behavior information for device owner, while limited guest's behavior information is limited. On the other hand, existing works do not

take multiple activity model into account, so that the identification mechanism may fail when user is in motion. In motion scenarios, some interacting features will be swamped by the motion from the perspective of sensory data, which greatly increases the difficulty of successful and precise identification. Therefore, the first challenge of this proposal is how to distinguish legitimate users from guests via limited information effectively and implicitly, even if the interacting features are partially swamped.

(2) Balancing among accuracy, delay, and energy:

Continuous implicit identification through motion sensors requires small delay and high accuracy. However, frequent observation and computation may cause unwanted energy consumption for the mobile device, especially when the current user is the owner or a guest is using an insensitive app, *e.g.*, playing a game. In addition, intrusion may happen when the sensors are off and the risk increases with the detection delays. Therefore, the second challenge of this proposal is to design a well formulated mechanism, which decides the observation timing to reduce energy consumption while guarantee the identification accuracy and short delay.

(3) Privacy appeal express:

In order to allow people to express their privacy desire and enforce mainstream online social network and PSPs to respect the desire, a healthy online photo sharing ecosystem is encouraged to be designed, implemented and evaluated. To achieve such purpose, I propose a privacy expression tag to be carried by people, which should be reliably detectable yet less noticeable. People is allowed to express their privacy desire through the ”Privacy.Tag” so that mainstream online social network and PSPs are enforced to respect the desire and promote a healthy online photo sharing ecosystem. When taking a photo, the privacy tag

should be localized to pinpoint the wearers and work with all cameras including legacy ones. The privacy tag must either contain the privacy policies directly or point to where the policies are.

(4) Respect privacy desire and grant privacy control:

When detecting a Privacy.Tag in the picture by PSPs and online social networks, the following challenge is how to respect the privacy desire and grant privacy control (*i.e.*, the control of photo's publicity scope) back to the user. I propose a reversible pattern guided obfuscation process to protect the faces in the photos, and the face is protected by the targeted user's public key retrieved from the tag owner. I would like to design a protocol to control the scope of a photo's publicity by controlling the dissemination of the key no matter who took and shared the photo. The proposed design should also save the PSPs from storing the original copies, but when providing evidence for law enforcement purposes the original photo shall be presented despite the criminal has wear the privacy tag. In addition, we also have to handle the scenario where multiple people in the same picture, so that determining each people's privacy desire desperately is becoming another important task.

(5) Guarantee quality loss of pirate video:

Modern mobile devices and smart devices are equipped with sophisticated cameras which are imitation of the human eyes. In order to prevent the video being captured with precise color, the system should decompose the color of the pixel in original video. However, chromatic decomposition will definitely affect the viewing experience. Therefore, because of persistence of vision, a pair of corresponding pixels having certain chromaticity combination could achieve the desired mix color. Since a video is a continuous serial of still image, a challenge here is how to decompose color in a random pattern so that the

continuous video could be different when being captured by cameras.

(6) Quality loss imperceivable to live audience:

When broadcasting video in public, the most challenging part of the system is to ensure the encoded illuminance flicker and color distortion are imperceivable to the legitimate viewers at first, and then become perceivable after a piracy procedure. In order to address this challenge, I will investigate the disparity between the human vision system and the camera system. The human eyes receives light illuminance and chromatic perturbations in a continuous but low-pass manner, while camera captures light as a discontinuous sampling system with a higher temporal resolution. Therefore, taking the continuous frame stream as a varying light signal with specific spatial and temporal color distortion, the challenge may be conquered by exploiting the information loss and distortion by camera shooting to look for opportunities.

CHAPTER 2: BACKGROUND AND RELATED WORKS

Since the full scope privacy preservation protocol consists of three main phases, it involves multiple techniques ranging from local access control to online privacy appeal expression, from visual encryption to public video ownership protection. In this chapter, I will introduce latest research problems and existing systems which are similar to but substantially different from my proposed problems.

2.1 Implicit User Identification and Authentication

The implicit user identification and authentication is one of the most important components in the proposed protocol, and it has been considered as a primary active defense mechanism [65]. The existing works on user identification and authentication by mobile devices mainly focus on both smartphones and tablets. Generally, there are two major categories of biometrics for user identification: physiology and behavioral biometrics [33].

Some physiological biometrics require special recognition components to collect users' physiological features, such as fingerprint, facial features, or voice, but they suffer high computational and energy cost. However, the error rate of these approaches are relatively high, for instance, the EER (Equal Error Rates) for facial recognition is around 28% [68], while the voice is approximately 5% [60]. Although fingerprint sensor has been equipped into some smartphones or tablets recently, the latest report reveals that the fingerprint scanner could also be breached easily [1].

For behavioral biometrics based schemes, some scientists employ user's usage features

on the phone (*e.g.*, texting or calling) or location pattern [97, 99], which, however, usually take a significant amount of time to determine the legitimacy of current user. Although some works have been conducted through typing behaviors or keystroke [38, 45, 58, 66, 74, 77, 113, 116], modeling typing behavior for individual on touch screen is inherently difficult. The motion sensors integrated in most modern smart devices have stimulated the research on user behavior detection, such as gait pattern recognition [46, 47, 69, 75], which usually have low true positive rate because of the diversity of gaits of people facing different types of surfaces, such as roads, snow, grass, *etc.*. Some researchers employ proximity to known devices as metric to identify the current role us user [63, 91].

Recently, researchers designed new identification methods by combining different biometrics. For instance, Clark *et al.* [37] proposed a framework to conduct continuous and transparent authentication by combining facial, voice and keystroke biometrics, while Crawford *et al.* [40] propose a multi-model system to conduct identification scheme. The latest framework is proposed as Itus [65], which offers researchers to improve the implicit authentication scheme dramatically by allowing developers to adopt these improvement by their own.

However, most of existing works could not provide transparent, responsive and continuous identification without intervening current operation in real time, while taking the balance the accuracy, delay as well as the energy consumption into account.

2.2 Privacy Appeal Expression and Reaction

Privacy preserving regarding users' online activities is a broad topic and have attracted extensive research attentions. Many existing works pay more attention to how the privacy is revealed when sharing photos or videos online [22, 23, 28, 29, 50, 67, 102]. However, this

dissertation mainly focuses on how to let user explicitly express their privacy desire and how the online social network and service providers should react to respect them accordingly, which is orthogonal to these problems.

Some privacy protection purposed mechanism are emerging, such as PriSurv [35], which utilizes RFID to control personal information disclose. Equipping near infrared LEDs to glasses, which emits invisible light but can be captured by camera, to convert hidden privacy appeal of not taking photos of me is invented [112]. These works require to work with either instrumented surveillance systems or with smart cameras, which increase extra cost, and is difficult to be spreaded easily. Although TagMeNot [16] alleviates the cost by using QR-tags, which allows people to express their privacy concern and calls for photo-takers to avoid publishing photos of them, they shift the burden to the photo-taker.

In addition, it is difficult to determine whether the privacy desire is respected or not by existing solutions. In our work, this dissertation outs emphasis on designing a systematic and automatic privacy protection solution involving mainstream online social photo sharing services without any assumption on cameras and human in the loop.

2.3 Image and Video Privacy Protection

Protecting privacy information in image is always a hot topic, especially when sharing images online is becoming increasingly popular [22, 23, 28, 29, 50, 67, 102]. Some efforts have been taken to achieve this purpose, such as concealing a person, blurring faces, masking and mosaicking the selected area of a image [5, 42, 83]. Some methods put emphasis on providing privacy preserving face recognition in a face photos database [93]. P3 [85] extracts and encrypts the most significant information of an image while preserving the rest in the public. Bo *et al.* [32] proposes a privacy expressing protocol, which requires

people to wear a Privacy.Tag to express their privacy desire and the photo sharing services to exert privacy protection by following users' policy expression. A number of creative methods [41, 43, 95] were proposed for protecting the privacy of objects in a video. For instance, [80] removes people's facial characteristics from video frame for privacy protection. [98] proposes that denaturing should not only involve content modification but also meta-data modification. Although existing techniques could offer certain level of privacy protection to the image or video themselves, it is difficult to control the propagation of unauthorized version or high quality version from pirate photo/video shooting.

2.4 Visual Cryptography

Visual cryptography [78] is designed as a simple but secure solution for image encryption, which exploiting Human Visual System to recognize a secret image from overlapping shares without any additional computational overhead required by traditional image cryptography schemes. Multiple algorithms [79, 84] have implemented to encrypt an image into another image. In addition, Rijmen *et al.* [90] expands each pixels of secret image into 2×2 blocks to encrypt color images, and Hou *et al.* [53] presents three different solutions for encrypting both gray and color images leveraging the halftone technology and color decomposition. Besides, Sozan *et al.* [20] designs a different approach by splitting an image into three different sub-images according to three primitive color components.

CHAPTER 3: PRIVACY PRESERVING ON SMART DEVICES: ACCESSING

For the purpose of protecting users' private information stored locally in the smart devices, in this dissertation, I would like to design a continuous authentication strategy (*SilentSense*) based on the pure software-based framework running on top of mobile system, which non-intrusively explores the behavior of users interacting with the devices and distinguishing the legitimate user from guest user or even attackers.

3.1 Motivation and Design Goal

The motivation of *SilentSense* roots from the growing privacy and security concerns facing the increasingly accumulated private data stored in the mobile devices. Similar to most of desktop systems, the mobile devices should also need to be locked or display different views to the non-legitimate users (*e.g.*, medical devices, finance applications or similar sensitive applications). Password/pattern-based authentication does not fully solve the problem because:

1. A password/pattern suffer from shoulder surfing attack;

2. Explicit authentication imposes inconvenience to user experience, which makes impatient users be reluctant to turn on the password/pattern-based protection in their devices;

3. The local personal data is no longer protected once the device is unlocked.

The last vulnerability is particularly critical because mobile devices usually need a period

of inactivity time before being locked, unless the legitimate user locks the device every time he stops using it. One simple solution is to let users explicitly lock the devices after using, but we believe a robust and safe authentication should not rely on users' action since users tend to forget, or the devices are delivered to guest users. Similarly, we do not consider that photo-based authentication is safe enough, since an attacker can simply use owner's photos to launch an impersonation attack, and it is also highly inconvenient since a user needs to take a photo every time when trying to unlock the device.

In conclusion, we believe that a reliable authentication system for mobile devices must have the following characteristics:

1. *Impersonation-proof*: the information/pattern used for authentication should be very difficult to replicate or imitate.
2. *Obliviousness*: the authentication process must be transparent to the users, which means the authentication must be conducted from users' regular usage on the devices.
3. *Continuity*: while someone is operating the device, the authentication should be conducted continuously to prevent an attacker from illegally accessing the device, or even to remove the necessity of the unlock.
4. *Quick in response*: the delay of the authentication should be short enough to reduce the risk of personal data being attacked.

The last property is introduced because of the obliviousness. Since the authentication is oblivious to users, the device will not take any action until the authentication fails, so if the delay is too long (*e.g.*, 1 minute), attackers have enough time to conduct simple attacks (*e.g.*, stealing sensitive information).

3.2 Main Idea and Challenges

The main idea of *SilentSense* is inspired by our preliminary experiment, which consists of two aspects: (1) each user, interacting with the device, follow their individual unique habits, and (2) such using habit is difficult to be imitated. Therefore, a feasible solution is to infer users' unique habits to construct an authentication model, and adaptively adjust and use the constructed model continuously when a user is interacting with the mobile device.

We notice that when users interact with the device by touching the screen, the whole device vibrates accordingly and such perturbation will be captured by motion sensors, including the accelerometer and gyroscope. In addition, the amplitude of such tiny perturbation depends on the user's holding gesture, the touching pressure and coordinate. Under this circumstance, the framework focuses on extracting features from the user's behavior, including both screen-touch events and the user's motion events, to build the discriminative patterns of individuals. Such behavior pattern and dynamics are much difficult to be imitated or attacked as these are often invisible. Besides, both our investigation and [64] show that people share phone with friends from time to time and the share frequency varies with the owners' social habit. The phone belonging to a more sociable owner, tends to have a higher probability to be shared with guest users, which may require a relatively high identification frequency, and vice versa. The social characteristic of the owner can help us to optimize the observation frequency to reduce the overall energy cost with a identification performance guarantee.

While the owner is using the phone, it is feasible to establish a behavior model through automatically learning. When interacting happens, the system evaluates the probability

of being the owner, and updates the evaluation with continuous monitoring to determine the user's identity silently and automatically. If the current user is a guest, the privacy protection mechanism will be triggered automatically to prevent the privacy leakage while maintaining the trustiness of the guest user. Based on the historical identification results, the social characteristic of the owner could be learned to help decide the observation frequency.

In order to design and implement the propose framework, the following technical challenges must be handled.

(1) Human behavior modeling:

Most of the existing works focus on the data they have collected, and build the authentication model with a SVM model. However, we further study when users actually interact with the device and how to determine the actual data source (from legitimate user or the guest), and found our preliminary experiment as well as existing surveys show that even the same user may have diverse behavior patterns when using the mobile devices (*e.g.*, gestures when holding the device, the touch behaviors for different applications). Facing such noisy and unpredictable user behaviors, constructing a reliable authentication model is a great challenge.

(2) Unsupervised Learning:

SilentSense does not ask for the ground truth dataset associated with the owner by asking him to perform several touch actions because this violates the obliviousness of our design. Then, conduct the unsupervised learning is a must.

(3) Feature Selection:

According to previous study and our own experiment, many features (over 30) could be

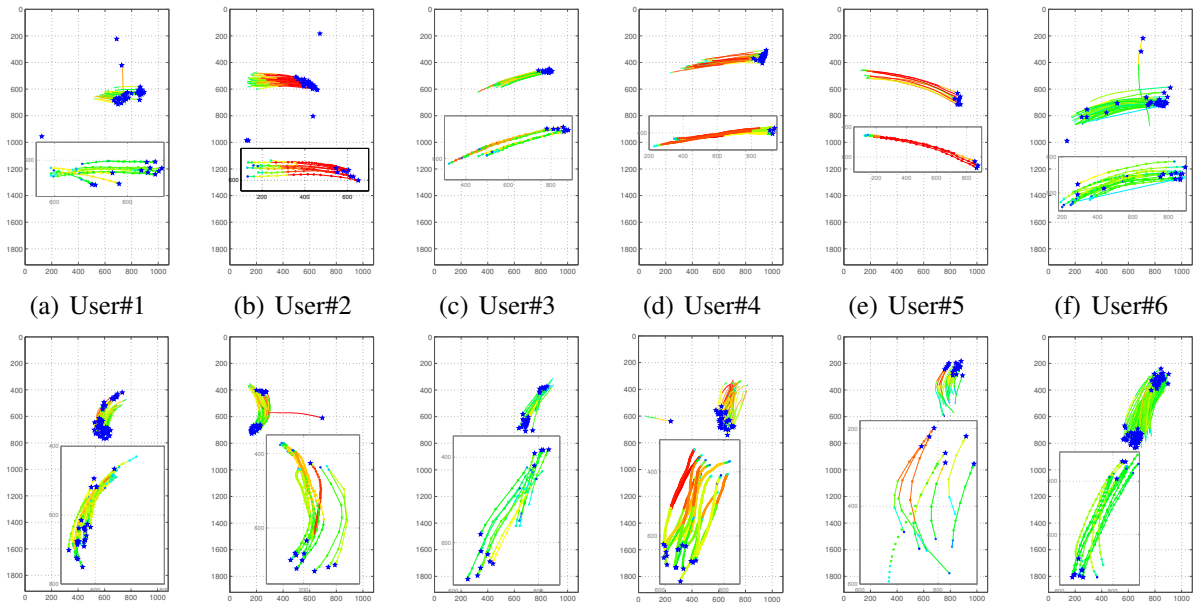
captured from one complete interaction. However, these may include noisy features which are not useful in the authentication, therefore we have to select the features that can result in high authentication accuracy.

(4) Continuous Authentication:

The continuous authentication imposes much greater computation overhead and complicated authentication strategy, and this becomes a critical challenge when the platform is a resource-bounded mobile device. To continuously determine whether someone is legitimate when he uses the mobile device, *SilentSense* continuously extracts and feeds action-related features to the authentication model and predicts whether this information comes from a legitimate user based on received information. The authentication model cannot guarantee 100% accuracy, therefore it needs to receive enough information before making a decision with a high accuracy. However, there exists a trade-off between the accuracy and the latency of the decision. If a decision is made too fast, it might be wrong due to the inadequate input data; if a decision is made with a very high accuracy, it may have waited too much time before making the decision. Both cases cause problems in the authentication because 1) the former one leads to inaccurate authentication, and 2) the latter one allows attackers launch attack before are stopped caught by the system. Therefore, a balanced must be found.

(5) Adaptiveness:

Since our authentication is based on users' behavior-related information, we need to adaptively adjust the authentication model because users' behaviors may change given a long period of time (*e.g.*, getting used to the new device, natural change), and the model becomes obsolete otherwise.



(a) User#1 (b) User#2 (c) User#3 (d) User#4 (e) User#5 (f) User#6

(g) User#1 (h) User#2 (i) User#3 (j) User#4 (k) User#5 (l) User#6

Figure 1: Stroke monitors of different users: browsing photos (a to f) and tweets (g to i).

3.3 Human Behavior Modeling

As interacting with the devices, various hidden information could be obtained to model human behavior, and the action is represented with a data instance with multitude of features.

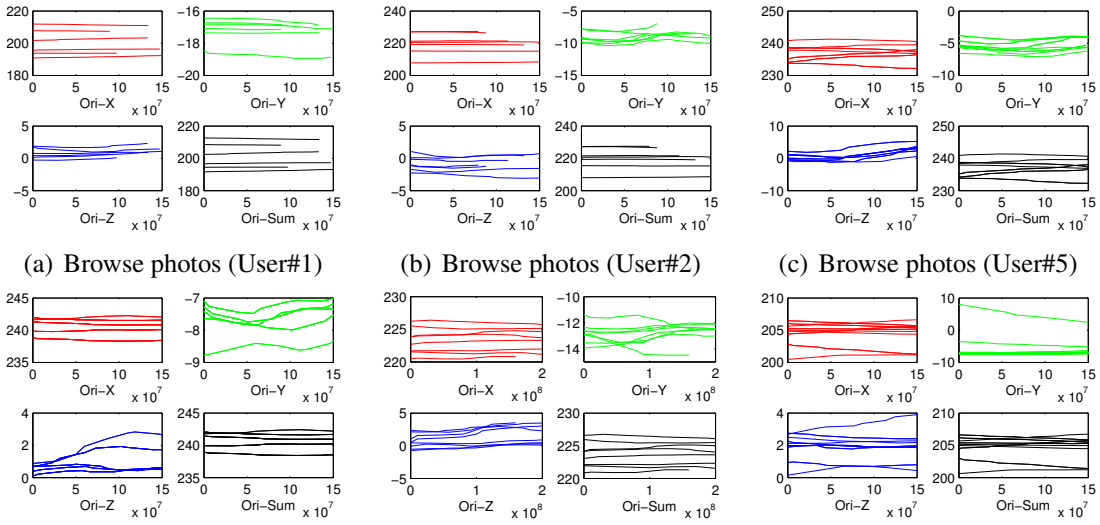
3.3.1 Feature of Touch Strokes

By analyzing different users' strokes on the touch screen(Fig. 1), we observed three categories of features:

1. Space features: direction, coordinates, bounding box and shape of the touch stroke;

2. Time feature: speed and duration of the touch action;

3. Pressure: the touch pressure.



(a) Browse photos (User#1) (b) Browse photos (User#2) (c) Browse photos (User#5)

(d) Browse tweets (User#1) (e) Browse tweets (User#2) (f) Browse tweets (User#5)

Figure 2: Phone orientation while different users are operating on the screen.

3.3.2 Free-Form Strokes

Our preliminary experiment indicates that the touch stokes for the same user show several similar patterns, while different users' stokes perform diverse shapes. Due to the variance of the stroke length and misaligned sample points, it is difficult to compare stokes by directly computing their similarity. On the other hand, we notice that a parametric Bézier curve models free-form smooth curves in the computer graphic area, which can be scaled indefinitely. The limited parameters of Bézier curve greatly reduce the data dimension but accurate enough to recover most shape information about the curve. Inspired by those work, we propose to approximately describe each stroke by a Bézier curve.

We first introduce the background knowledge of the Bézier curve briefly. A Bézier curve is defined by a set of control points $\mathbf{P} = \{\mathbf{P}_0, \cdots, \mathbf{P}_n\}$, where n is called its order (n = 1 for linear, 2 for quadratic, 3 for cubic etc.) The first and last control points are always the end points of the curve. Figure 3 shows an example of two different form cubic Bézier curves.

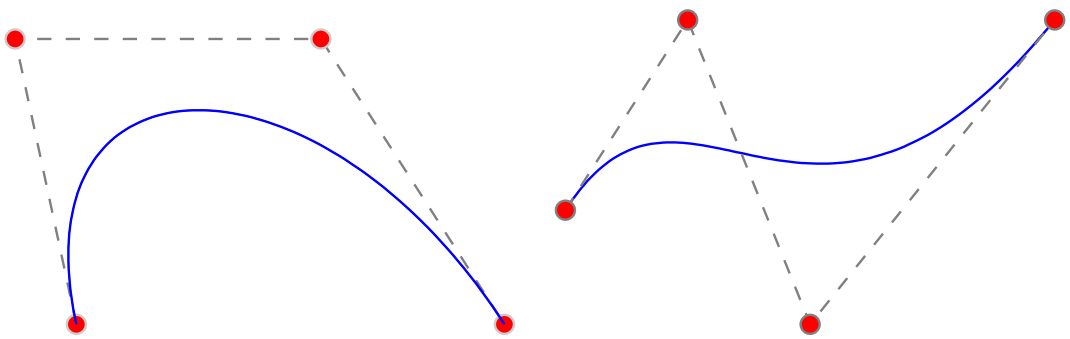


Figure 3: Example of cubic Bézier curves and their control points denoted by red dots.

A Bézier curve can be represented as an equation

$$p(t) = \sum_{i=0}^{n} B_i^n(t)\mathbf{P}_i. \tag{1}$$

where

$$B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i$$

Next, we need to fit the discrete sample points of a stoke with a Bézier curve. This problem is formed as fitting a given ordered set of data with a cubic Bézier curve in the total least squares sense: given an ordered set of points $D = \{d_i, i = 1, 2, \cdots, m\}$, find a set of control points $\mathbf{P} = \{\mathbf{P}_j, j = 1, 2, \cdots, n\}$ and a vector t of nodes, $0 \leq t_1 \leq t_2 \leq \cdots \leq t_m \leq 1$ that minimize

$$||B(t)\mathbf{P} - D||_F \tag{2}$$

Here the Frobenius norm of matrix $A \in R^{m\times n}$ is given by

$$||A||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j}^2}. \tag{3}$$

This minimization problem is a nonlinear least squares problem, and in our system we solve it using the Gauss-Newton method. Note that, for the solution $\mathbf{P}, t$, P is the feature

which determines the shape of the curve.

In our system design, we find that most strokes' shapes are not complicated, as depicted in Fig. 1. For representing touch strokes, order-3 (cubic) Bézier curve is sufficient to describe their shapes. Fig. 4 gives some examples of Bézier fitting results. It shows cubic Bézier curves form the touch stroke perfectly, and variable length strokes are represented by 4 points.

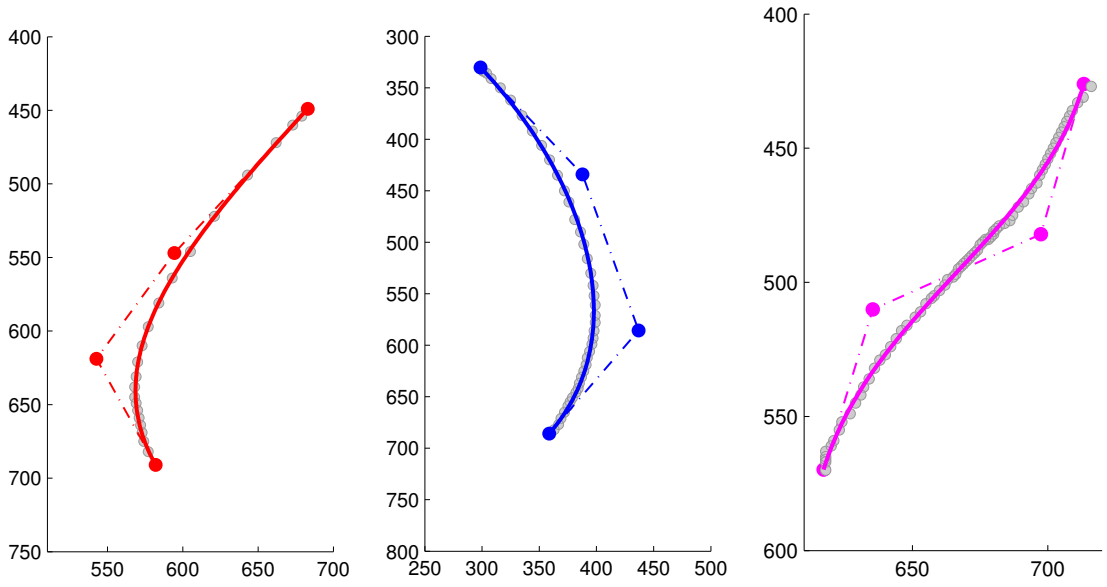
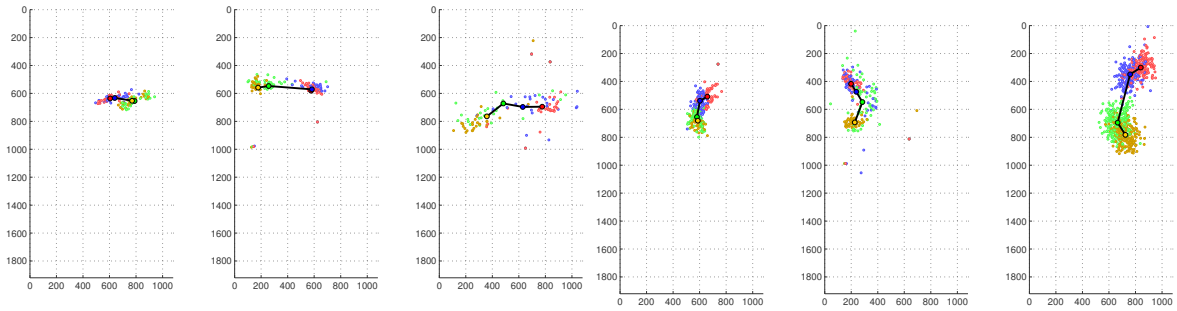


Figure 4: Example of cubic Bézier curves fitting results for three users' touch strokes.

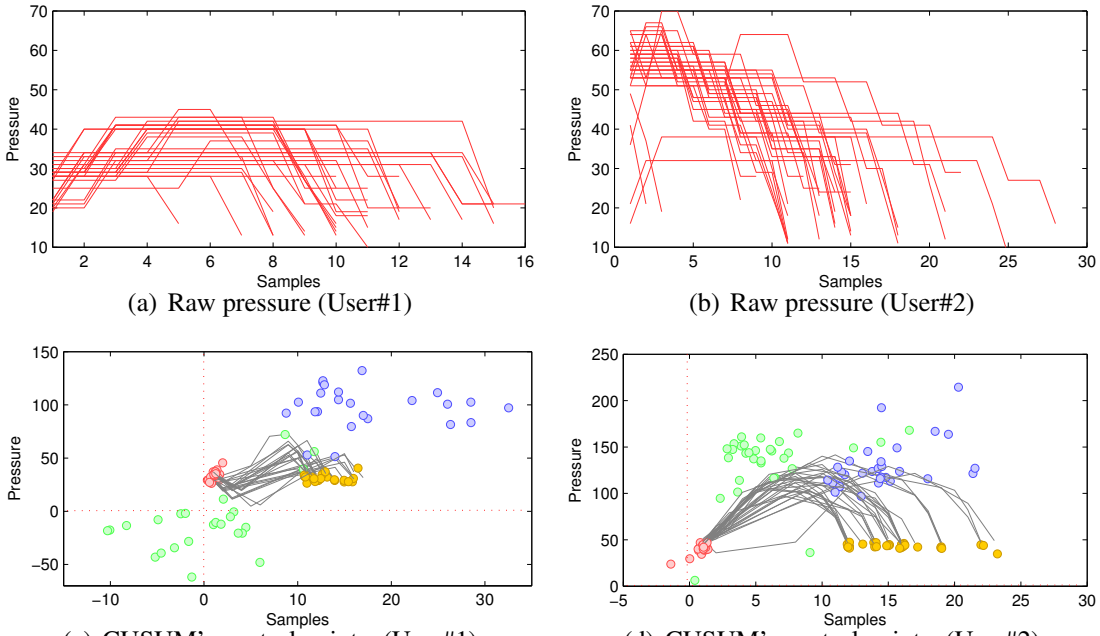
Until now, we have modeled users' touch strokes by four control points of cubic Bézier curves, which implies the coordinates, direction and shape of strokes simultaneously. The dimension of the curve is also reduced from hundreds of sampled points to lower and fixed dimension (4-D) with little sacrifice of shape information. We compute the control points of different users' stokes, and Fig.5 depicts some of these results. These results intuitively show us that the distribution of the four control points of one user's strokes follows some patterns and the distribution varies among users, which shows the consistency and diversity.



(a) Browse photos (User#1) (b) Browse photos (User#2) (c) Browse photos (User#3) (d) Browse tweets (User#1) (e) Browse tweets (User#2) (f) Browse tweets (User#3)

Figure 5: Cubic control points of strokes of different users.

3.3.3 Pressure



(a) Raw pressure (User#1) (b) Raw pressure (User#2)

(c) CUSUM's control points. (User#1) (d) CUSUM's control points. (User#2)

Figure 6: Example of pressure change analysis.

Pressure changes with the stroke and its mean and variance are used as features by exiting work. However, as shown in Fig. 6(a) and 6(b), commercial smart devices can only sense limited discrete pressure levels while the user is touching the screen, which determines that the magnitude of mean and variance of pressure could be too limited to characterize diverse

users. We notice that the change trend of the pressure, intuitively *e.g.* 'first hard, then gently', is also user-dependent, which provides larger capability to distinguish different users. However, with the raw pressure data (Fig. 6(a) and 6(b)), it is non-trivial to characterize the subtle difference of change trends.

In this work, to better characterize the changes, we begin with a specially designed cumulative sum chart (CUSUM). CUSUM chart is capable of detecting subtle changes and robust to outliers. Given a sequence of input data points $D = \{d_1, d_2, \cdots, d_m\}$, the CUSUM chart $C = \{c_0, c_1, \cdots, c_m\}$ is constructed in the following way:

1. compute the average $\bar{d} = \sum_i d_i/m$;
2. start the cumulative sum at $\bar{d}$, *i.e.*, $c_0 = \bar{d}$;
3. compute the other cumulative sums $c_i = c_{i-1} + (d_i - \bar{d})$.

Instead of cumulative sum of the raw data, CUSUM computes the cumulative sum of differences between the values and the average, and the cumulative sum also ends at the average. The grey lines in Fig. 6(c) and 6(d) are CUSUM charts of the raw pressures. In the CUSUM chart, a segment with an upward slope indicates a period where the values tend to be above average, and the change of the gradient indicates the change trend of the raw data, vice versa. When the data changes randomly, the CUSUM chart goes up and down near the average. After we get the CUSUM chart of raw data, we fit it with a cubic Bézier curve, which characterizes its shape. Specially, for the first and the last control points, their values are $(0, \bar{d})$ and $(m, \bar{d})$ respectively. Then we use the fitting curve's four control points as the pressure feature, as shown in Fig. 6(c) and 6(d), which implies the mean and subtle change trend of pressure.

3.3.4 Time Feature of Strokes

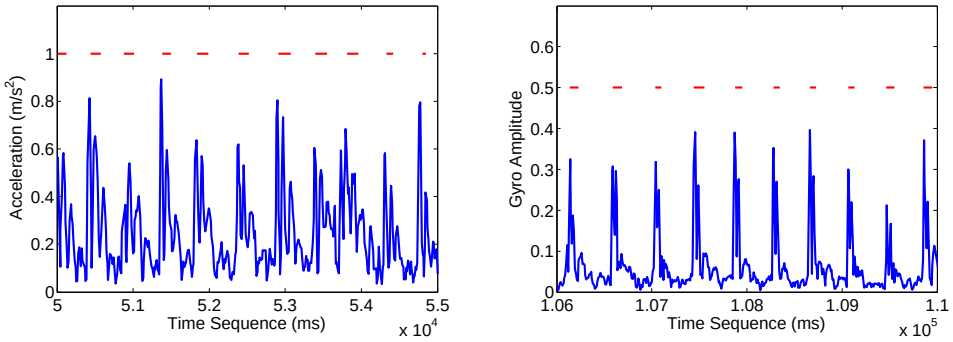
People move their fingers on screen at different speeds. Existing works use mean and variance of each stroke's speed as well as its duration as time-domain features, but discard the rich information of the speed change trends. Similar to process pressure, we extract the speed feature with control points of CUSUM chart.

3.3.5 Motion Sensors

There are three main patterns of interacting with touch-screen based devices (*tap*, *e.g.*, texting, clicking item, *scroll*, *e.g.*, browsing mails and tweets, and *fling*, *e.g.*, reading e-books). Different type of strokes usually have various touch features which lead to different device reactions. For this goal, we utilize the embedded motion sensors in mobile devices to explore the device reaction when being touched by the three types of strokes.

We start the analysis the action of tapping. When user touches the screen by tapping, the system API provides the the tapping coordinate, timestamp and duration. Meanwhile, the device reacts by tiny vibration, and the vibration amplitude is reflected on the readings from both accelerometer and gyroscope. We quantify the amplitude of vibration from the accelerations along three device axes (X, Y, Z), and use the $F_{tap} = \sqrt{{LA_x}^2 + {LA_y}^2 + {LA_z}^2}$ to represent the summation of acceleration vector in the space, where LA_x, LA_y, LA_z indicate the linear acceleration in the device system respectively. Another valuable reaction feature is the vector of angular velocity obtained from the gyroscope, denoted by $AV_{tap} = \sqrt{{AV_x}^2 + {AV_y}^2 + {AV_z}^2}$. This feature represents the position variation of the device in the space when touched by a user.

Figure 7 illustrates the reaction of the mobile device when tapping event occurs while the user is sitting or standing still. In both sub-figures, the red line segments represents

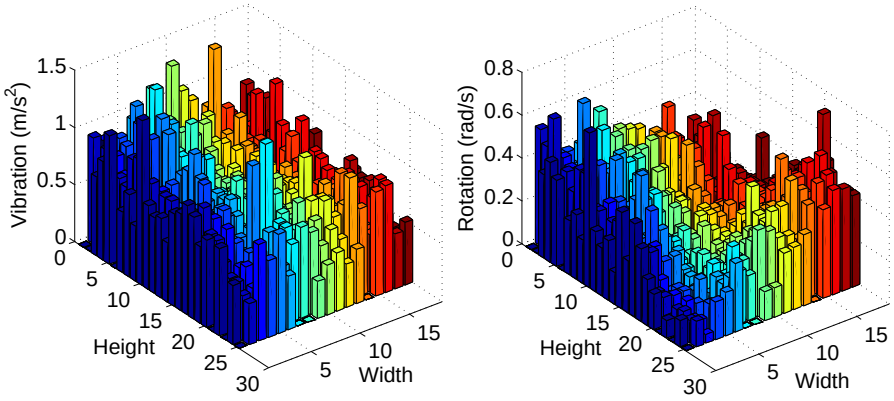


(a) Accelerometer in tap (b) Gyroscope in tap

Figure 7: The reaction of the device when tapped.

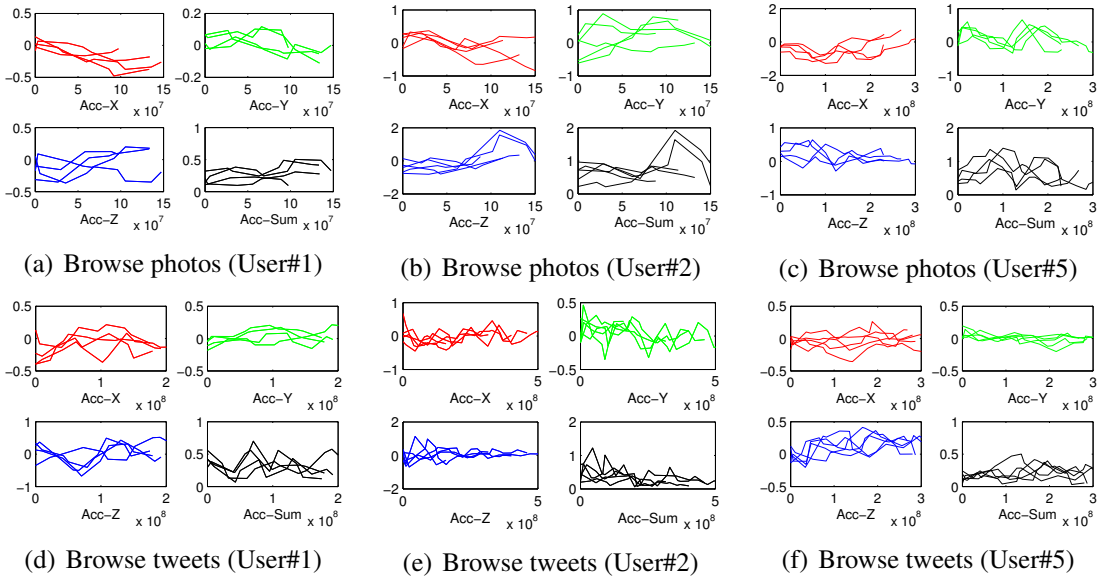
the occurrences of tapping, and the length of each segment corresponds to the duration of the tap event. Tapping on the screen will cause jumping on the sensory data. However, the sensory data contains both intrinsic noise and measurement error (User cannot hold the device in absolute still), we calculate the mean perturbation of both acceleration and rotation within each stroke as first-order features.

We conduct a long period experiments to measure the vibration and rotation of the device with various touch coordinates. We separate the touch screen into 25×15 small grids and calculate the mean vibration and mean rotation caused by tapping from each user for within grid-cell. Figure 8 shows the statistic device reactions from one of the users, who used to hold the lower part of the device by left hand, *i.e.* the supporting point is near the left bottom, and taps the device by right index finger. The experiments results show some interesting observations: (1) the amplitude of vibration and rotation depend on how the user holds the device: the father the coordinate from the holding point, the larger the vibration and rotation will be; (2) the changing trend of the vibration is obvious, leading to the possible holding position (which is also a behavior biometrics).



(a) Vibration in tap (b) Rotation in tap

Figure 8: The distribution of both vibration and rotation on touchscreen under a given holding gesture.

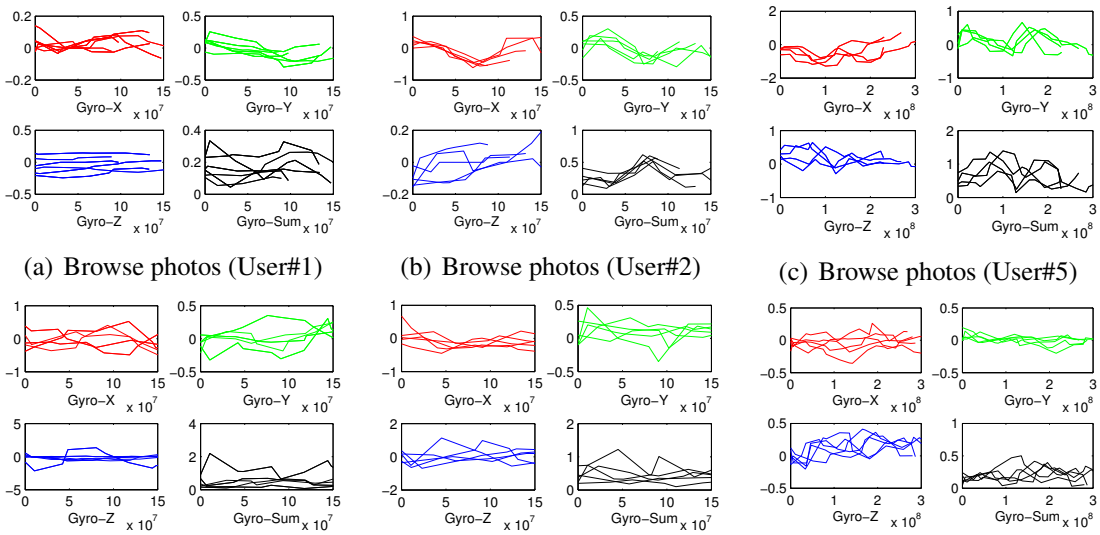


(a) Browse photos (User#1) (b) Browse photos (User#2) (c) Browse photos (User#5)

(d) Browse tweets (User#1) (e) Browse tweets (User#2) (f) Browse tweets (User#5)

Figure 9: Acceleration while different users are operating on the screen.

When interacting on the screen, the change of linear and angular accelerations show some patterns, but the orientations of the device tend to be stable while actions, as shown in Fig. 9 and Fig. 10. Therefore, we use control points of CUSUM charts to characterize accelerations while use mean value for orientation.



(a) Browse photos (User#1) (b) Browse photos (User#2) (c) Browse photos (User#5)

(d) Browse tweets (User#1) (e) Browse tweets (User#2) (f) Browse tweets (User#5)

Figure 10: Rotation from gyroscope when different users are touching the screen.

3.3.6 Second-order Features

So far, we have presented only raw data and first-order features. Previous works usually use the raw data and some of the first-order features to compose the feature vector of user's action. However, this simple detection does not achieve high accuracy, and our preliminary experiments also show that the first-order features are not sufficient to effectively distinguish owners from others. This is because the SVM is a linear classifier, and the action-related data cannot be shattered due to the low dimension of the data instances (*i.e.*, not enough features), and we need more features to shatter the data.

To solve this problem, we further extract the *second-order feature* which characterizes the relationship between or among the first-order features. Since the kernel function of the SVM shows the pairwise mappings between each dimension of the feature space, given a set of data instances (*i.e.*, set of points in the space), we adjust the parameters until we get the best kernel function which delivers the best distinguishability. In this way, the second-

order features, *i.e.*, the relationship of the features, are implicitly represented by the SVM we choose, and we can exploit it to distinguish owners' clusters from others'.

In conclusion, we can extract the features in Table 1 from the touch actions of users as in Fig. 11.

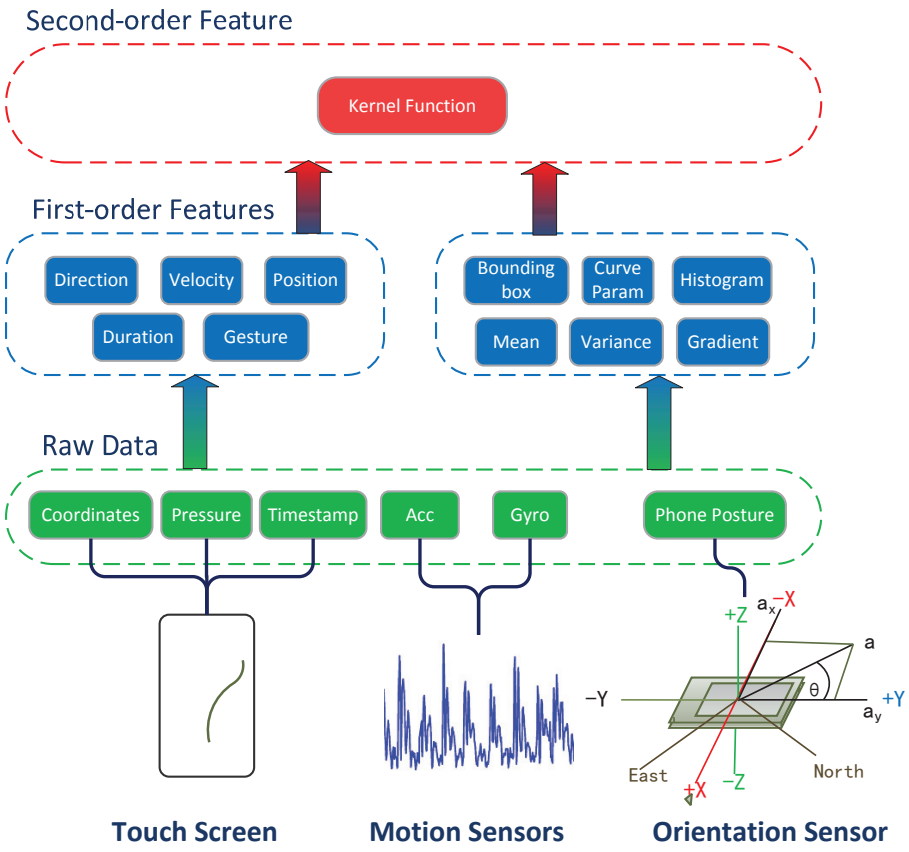


Figure 11: Multi-leveled features

3.4 Construction Authentication Model

Above features are used to build an authentication model, but in *SilentSense*, the ground truth dataset is not available for the model construction because the authentication is no longer oblivious if it forces users to honestly tell whether they are legitimate and then let them provide their behavior-related data. Therefore, *unsupervised learning* should be conducted if we want to train an authentication model on top of numerous unlabelled time-

Table 1: Collected Data & Features

Raw Data	
Touch Strokes	Coordinates, pressure and timestamps
Device Reaction	Acceleration, angular velocity and device orientation
First-order Features	
Stroke Shape	Bézier control points of the stroke
Stroke Characteristics	Bounding Box, duration, cumulative sum of pressure and velocity
Statistics	Mean, variance of stroke pressure, acceleration and angular velocity
Second-order Features	
Relationship among Features	Implicitly expressed with the kernel function of SVM

series data (whether they are from a legitimate user is unknown). Besides, not all features extracted from the device are useful in building the authentication model, and some data may even lower the accuracy due to the noise which hides the weak signal in the data. Therefore *feature selection* is needed to select the significant features.

3.4.1 Unsupervised Learning

Typical unsupervised learning can find the label structures among unlabelled data (*e.g.*, which data instances belongs to the same group), but it is impossible to assign correct labels without certain priori information. To correctly find the owners' input data (*i.e.*, action-generated features) among numerous unlabelled dataset, We empirically assume that the owner is the one who uses the device for majority of the time, and we conduct our unsupervised learning based on this assumption. According to our preliminary experiments, there is a huge gap between an owner's cluster and a non-owner's cluster in terms of their size. That is, the owner's clusters are usually much greater than non-owners' clusters. Therefore, we use OPTICS [26] to cluster the data instances and sort the sizes of the clusters in non-increasing order. Then, we compare all adjacent clusters to find the first difference greater than a threshold θ_{gap}, and consider all clusters in the front as owners' clusters.

Since there are multiple clusters for a single user, we first train a SVM for each clus-

ter. Then, for each time *SilentSense* observes an input data (*i.e.*, action-related features), it tries to classify it with all SVMs that it has. If it is surely from an unauthorized user, *SilentSense* immediately locks the device; if it is surely from an owner, *SilentSense* does nothing; otherwise, *SilentSense* cumulates the information until it achieves enough confidence about the authentication (Section 3.5). Whether the user is owner or not, our system stores the input data to adaptively adjust our model later (Section 3.6).

Then, each time *SilentSense* observes an input data (*i.e.*, action-related features), it first finds its nearest cluster to determine whether the action is made from the owner.

3.4.2 Feature Selection

In fact, not all features can be used in the above clustering due to the noisy data and computational overhead. Therefore, features which may negatively affect accuracy must be excluded in advance to the model training.

To positively contribute to the authentication, a feature has to be both *consistent over time* and *diverse among users*. A feature is consistent over the time domain if the feature's characteristic or value does not change much along the time for the same user, and a feature is diverse among users if the feature's characteristic or value varies greatly among users.

To measure the consistency and the diversity of each feature, we use OPTICS [26] to cluster the data by looking at each feature at once, and we use the maximum radius over all clusters r_{max} as the metric to measure the consistency of the feature, and the minimum pair-wise distance d_{min} over all pairs of clusters as the metric to measure the diversity of the feature. Then, we use d_{min}/r_{max} to evaluate the quality of the feature and uses the top-k features in our training, and include the features with high quality in the training, who are denoted as *significant features*.

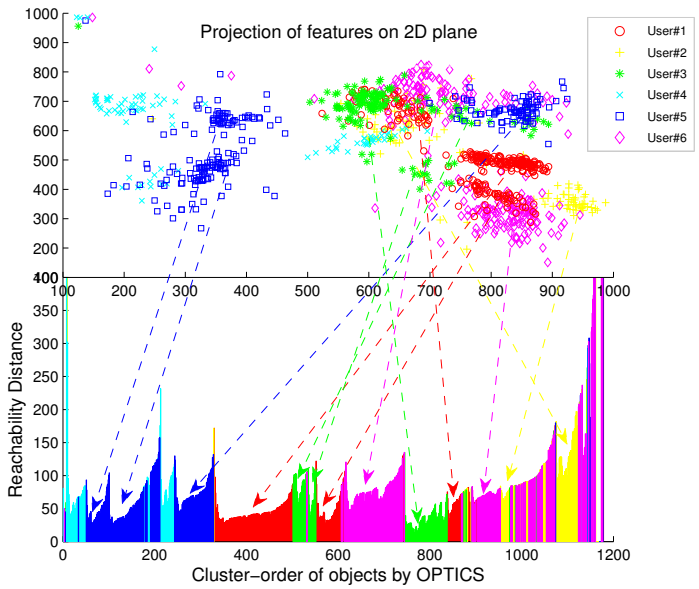
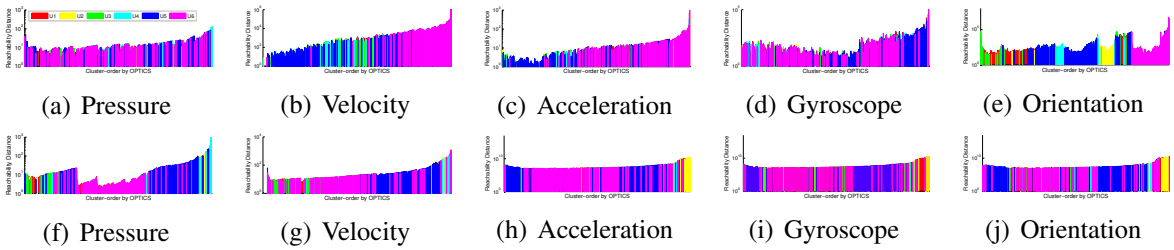


Figure 12: Clustering results of 6 users' strokes by control points using OPTICS with MinPts=5. Note that the upper figure is only two-dimension projection of the 8-D clustering space.

For example, Fig. 12 depicts the cluster radius of each cluster from OPTICS when clustering the data instances with their control points of strokes. The OPTICS algorithm generates an ordering of the input data points and corresponding reachability distance. The cluster-ordering of the data set can be represented and understood graphically. As depicted in the lower figure in Fig. 12, the clustering structure is presented by plotting the reachability distance for each data point in the cluster-ordering, where each "Gaussian bump" indicates a cluster. Fig. 12 presents about 10 "Gaussian bumps", each of which is pointed by an arrow from its corresponding cluster of data points in the upper figure. By showing clustering structure, it is obvious that control points of stroke is a good feature to use. Although d_{min} is small, r_{max} is also very small. The final significant features we chose based on the quality function d_{min}/r_{max} are: control points of the strokes, control points of the pressure's CUSUM chart and the mean & variance of the orientation. Fig. 13 shows the



(a) Pressure (b) Velocity (c) Acceleration (d) Gyroscope (e) Orientation

(f) Pressure (g) Velocity (h) Acceleration (i) Gyroscope (j) Orientation

Figure 13: Illustration consistency and diversity of different features using cluster-ordering by OPTICS.

consistency and diversity results of other behavior-related features, first five figures represent both mean value and variance for different interacting parameters, while the next five illustrates the control points for same parameters.. Intuitively, control points of pressure and mean& variance of orientation are good features.

3.5 Continuous Authentication

As aforementioned, a trade-off exists between the decision accuracy and the latency. However, the decision returned from SVM given an unknown data instance has certain probability to be a wrong decision. More observations lead to higher accuracy but longer latency, and fewer observations lead to shorter latency but lower accuracy. Therefore, we present the following decision strategy on top of our predictive model. Given an observed behavior-related information I_i at time i, the SVM provides a prediction D_i on whether the action at time i belongs to the owner (*i.e.*, D_i =T or F). We define $\varepsilon(D_i)$ as the *confidence* of this decision (*e.g.*, relevant confidence value provided by the clustering method), and define the following as the cumulative *final confidence* given a sequence of consistent decision $D_{1-k} = \{D_1, D_2, \cdots, D_k\}$:

$$C(D_{1-k}) = 1 - \left(\prod_{i=1}^{k} (1 - \varepsilon(D_i)) \right)$$

which indicates the final probability that the consistent decisions D_{1-k} are correct. If the final confidence is above a threshold, *SilentSense* finally accepts the decision. Note that an inconsistent judgement will interrupt the sequence, and the conclusion confidence needs to be cumulated from scratch.

Up till now, the decision delay for a conclusion is defined as the number of observations (*i.e.*, actions) taken to achieve this conclusion. With the number of observation increases, the decision becomes increasingly confident to provide a correct conclusion, meanwhile the delay will increase. The number of observations needed for a decision is affected by the confidence threshold θ_{conf}. One can set up different threshold values for different devices or applications based on the response time demand. High θ_{conf} leads to long response time and vice versa.

Furthermore, to reduce unnecessary energy consumption, we configure a 'duty cycle' of the authentication. That is, if the received information is predicted to be from a legitimate user, *SilentSense* stops authentication for the predefined time t_{stop} (if from an unauthorized user, device is just locked). The cycle t_{stop} may be set differently for different applications based on their privacy demand.

3.6 Adaptive Authentication

Besides, as aforementioned, users' behaviors may present different patterns when given a long period of time (*e.g.*, getting used to a new device, natural change). To capture and adapt to such behavior changes, *SilentSense* configures a time window t_{chg} and keeps the authentication model fresh by 1) removing old data from the labelled data outside the time window, and 2) adding the new data into the labelled dataset after its label (*i.e.*, legitimate or not) is predicted.

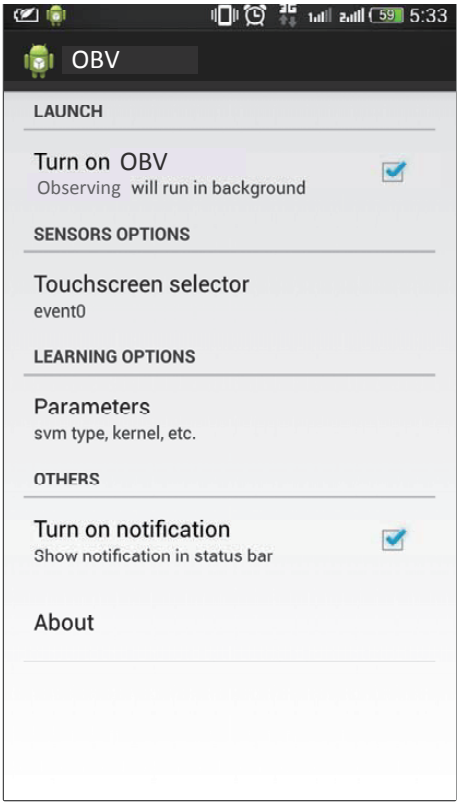


Figure 14: Screenshot of our system's configuration interface.

In theory, the feature selection must be repeated to build a new authentication model for the fresh data in the time window. However, it is too computationally intensive to keep the authentication model fresh all the time, and furthermore there is no such need since human behavior changes gradually and slowly. Therefore, in practice, we give a long period of interval (*e.g.*, 24 hours) between the updates, or we can also choose to update the model only when the mobile device is being recharged and inactive. For each update, we let *SilentSense* remove least recent behavior-related data outside the time window t_{chg}, and conduct the feature selection and authentication model building to get a new model.

3.7 Evaluation

3.7.1 System Implementation

We implement *SilentSense* as a service, running at the background of the Android phone. The service mainly captures the touching event from the system API as well as the reaction features of the smartphone from the integrated motion sensors. Then, we implement our Bezier curve fitting and CUSUM chart generation algorithm by JAVA to extract features out of raw data. For feature selection, the OPTICS clustering is realized based on the implementation of ELKI [21]. We develop the SVM model training and user identification component using LIBSVM [34].

3.7.2 Evaluation Configuration

In our evaluation, we employ our system in Android based HTC EVO 3D, HTC One and Samsung Galaxy S3 respectively with root privilege. We assemble 50 volunteers, and label 10 of them as owners to one of the Android phones. They own the phone for at least one day. During their ownership, they can use any application in the phone (*e.g.*, mail, photo album and social networking apps), and operate the phone freely. The rest 40 users are required to act as guests who borrow the phone from 10 owners and use it for several minutes from time to time. *SilentSense*records all users' touching events and phone reactions with afterwards marked name labels. Note that, the labels only serve as ground-truth to evaluate the accuracy of *SilentSense*, and they will not be used for user model training and identification.

We extract 11 different features for each action from the raw data, including

- 4 control points of each stroke curve (denoted as Sc);
- 4 control points of pressure CUSUM chart (denoted as Pc);

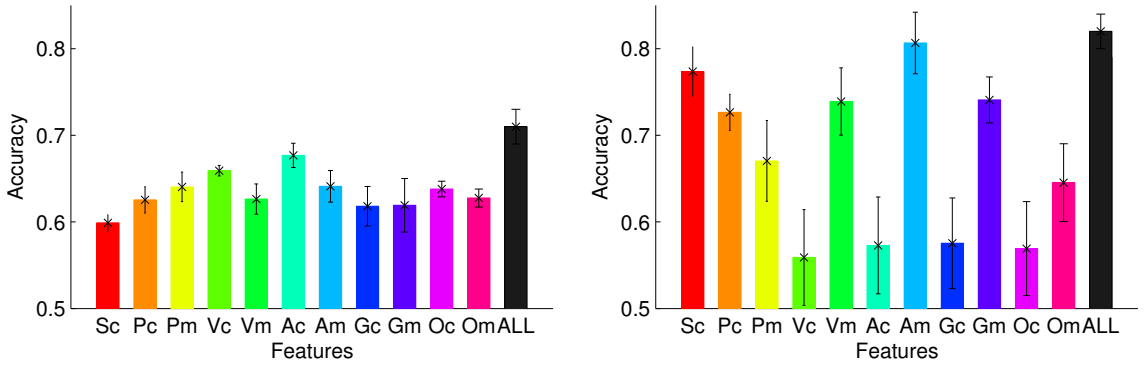
- mean and variance of pressure (denoted as Pm);
- 4 control points of velocity CUSUM chart (denoted as Vc);
- mean and variance of velocity (denoted as Vm);
- 4 control points of acceleration CUSUM chart (denoted as Ac);
- mean and variance of acceleration (denoted as Am);
- 4 control points of angular acceleration CUSUM chart (denoted as Gc));
- mean and variance of angular acceleration (denoted as Gm);
- 4 control points of orientation CUSUM chart (denoted as Oc);
- mean and variance of orientation (denoted as Om).

In our experiment results, for the abbreviations of features, 'S', 'P', 'V', 'A', 'G', 'O' stand for stroke, pressure, velocity, acceleration, gyro and orientation respecively; 'c' stands for control points and 'm' stands for the tuple of mean and variance. Here 'All' denotes the feature which is built by compounding all these features.

Using extracted features, actions are clustered by density. Clusters with larger action number are considered as the owner's data, and other data are considered as guests'. Then SVM models of owners are trained based on the clustering results.

3.7.3 Identification by One-class SVM Model

First, we evaluate the identification accuracy based on different features using one-class SVM models. Here "all" means to combine all features to form a compound feature of much higher dimension. By each type of feature, user actions are clustered, and only the clusters which are considered as the owner's are used to train one-class SVM models of the owner. Then these models are used for identification, which predicts the input action as "owner" or "guest". Figure 15 presents the identification accuracy statistics of all owners'



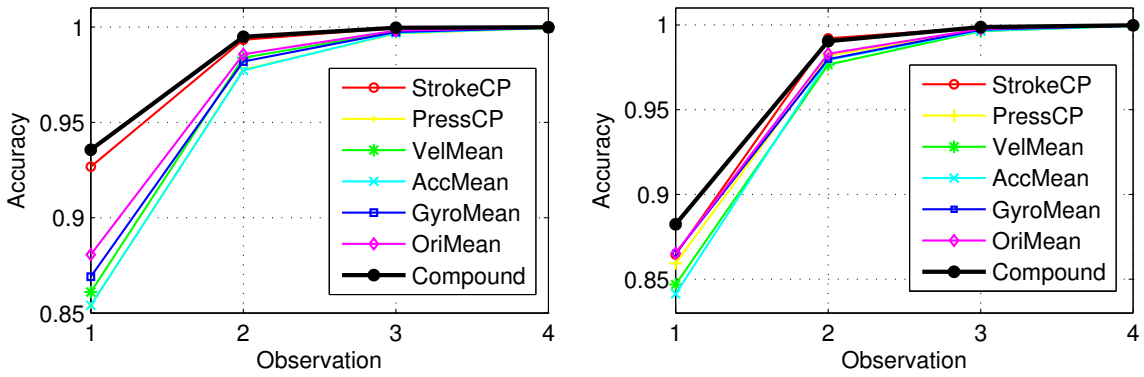
(a) Owner accuracy (b) Guest accuracy

Figure 15: Accuracy by one-class SVM model using different features.

models. For each model, there are about 250 input actions from its owner and more than 40 guests. The result inspires us to improve our system design in two aspects: (1) 11 features show different performance to identify owner and guest. Overall, control points of strokes and pressure, mean and variance of velocity, acceleration, angular acceleration and orientation outperform other features. This experiment result is consistence with our theoretical analysis for feature selection. We select these 6 features and compound them as the feature of users' actions in *SilentSense*. (2) With one-class SVM model, the identification accuracy for each action is not high enough. Using a single feature, the accuracy is below 70% for owners' actions and below 80% for guests' actions. Even using compound features, the accuracy is 70.3% and 81.5% for owner and guests respectively (black bars in Figure 15). To improve the accuracy, we employe multi-class SVM in *SilentSense*.

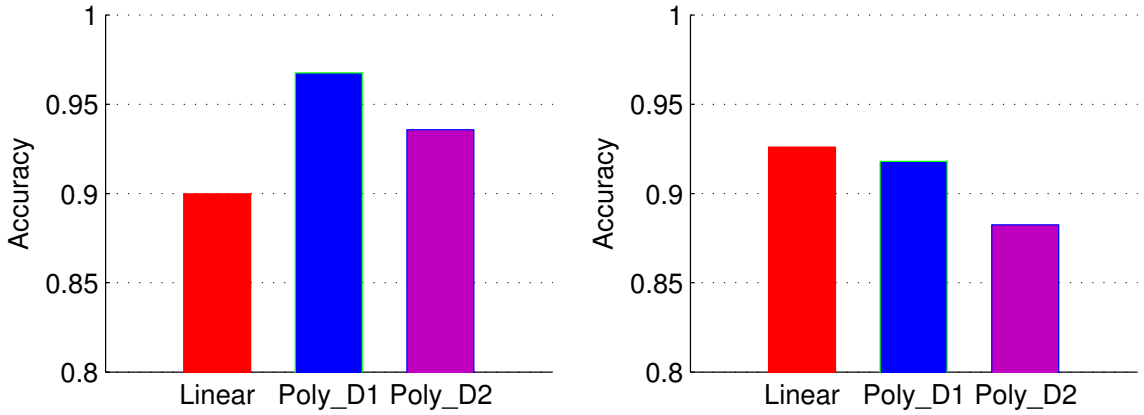
3.7.4 Identification with Multi-class SVM Model

Treating guests' data as one class and each of the owner's clusters as a class, we build multi-class SVM models for owners. An action is considered as the owner's if it is classified into any of the owners' classes, otherwise it is the guests'. And the multi-class SVM model also produces a probability to represent certainty for each prediction. Fig. 16 de-



(a) Owner accuracy (b) Guest accuracy

Figure 16: Accuracy by multi-class SVM model using different features.

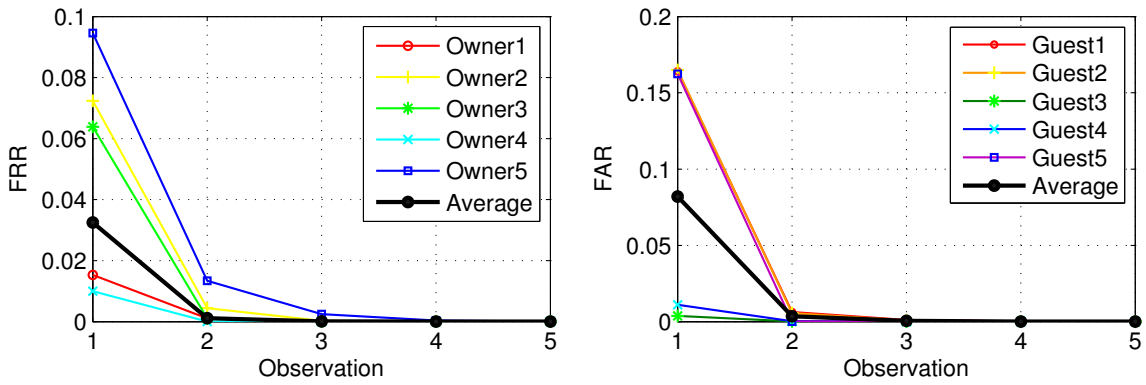


(a) Owner accuracy (b) Guest accuracy

Figure 17: Accuracy by two class SVM model using different kernel functions.

picts the identification accuracy with respect to number of observed actions using different features. Using multi-class SVM model, the ranking of each feature's identification ability remains similar to that using one-class SVM model. But compared to one-class model, the improvement of accuracy is significant. Using compound features, the average accuracy with one observation raises from 70.3% to 93.5% for the owner, and from 81.5% to 88.2% to guests. And the accuracy quickly reaches 100% within 4 observations for both owner and guest.

To achieve better performance, we also explore the performance change by using differ-

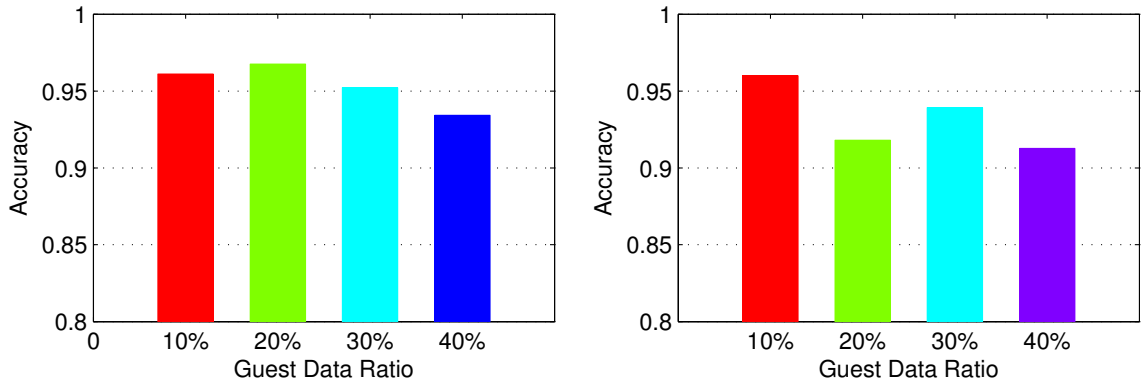


(a) FRR (b) FAR

Figure 18: FAR and FRR by different number of actions observed.

ent kernel functions for SVM model. We exam polynomial kernel functions with different degree in case that the compound features are unseparated in the linear space. Fig. 17 presents the experiments result. For identifying the owner, degree-2 polynomial kernel function performs best, while linear kernel function outperforms it a little bit to identify guests. Overall, degree-2 polynomial kernel function provides better accuracy.

Based on our evaluation, we use the compound feature (control points of stroke and pressure, and mean and variance of velocity, acceleration, angular acceleration and orientation) and degree-2 polynomial kernel function when deploying and implementation *SilentSense*in real application. We exam the False Acceptance Ratio (FAR), and False Rejection Ratio (FRR) of identification with different number of observations. Here the FAR is defined as the ratio of the number of identifications misjudging a guest as an owner over the total number of guest actions; and FRR is defined as the ratio of the number of identifications misjudging the owner as a guest over the total number of owner actions. The results of 50 users show surprisingly good results, as illustrated in Figure 18. The mean FRR of identification by one observation is 2.7% and it is reduced to below 0.1% after only



(a) Owner accuracy (b) Guest accuracy

Figure 19: Identity accuracy using models with different ratio of guest data.

observing about 2 actions. With about 3 observations the FRR achieves 0. Similarly, the mean FAR of identification by one observation is 7.8%. The FAR is reduced to below 0.1% after observing about 3 actions and with about 4 observations the FAR achieves 0. The experiments result show great efficiency and accuracy of *SilentSense* based on the extracted compound features and multi-class SVM models.

To deal with different owner and guest data size, we exam the performance of *SilentSense* with respect to the ratio of the guest data in the training data. Fig. 19 present the result, which shows that as long as the size of the guests' data is smaller than the size of the owner's data, which is true for most cases, the ratio only has a slight effect on the accuracy.

We also evaluate the performance of *SilentSense* when there is only a small set of data collected from users. In our test, we have collected training data with 20 strokes of owner and a few stroke from guest users (here we do not know the correct label for each data), the FRR is 8.2% with one observation and reduces to below 1% after two observing 2 new strokes. With more than 3 strokes, FRR reduces to 0. Similarly, the FAR is 7% for the first observation and reduces to below 1% after two observation. With more than 3 strokes,

FAR reduces to 0.

3.7.5 Mobile Users

In this part, we evaluate the performance of *SilentSense*in a different scenario, that users interact with smartphone while walking. Recall that, the vibration and rotation reaction features caused by touch are no longer feasible due to the large movement cased by walking. In this case, if only touch features (coordinate, pressure and duration) are used, although the FAR reduces to 0% after only 2 steps, the FRR is high as 18% even after 4 steps. In the dynamic scenario, we extract 4 walking features, including vertical displacement, step duration, mean and standard deviation of the horizontal acceleration, from the filtered accelerations. First, we explore the discriminative of walking features. We found that the walking pattern varies greatly for different users, which give us an opportunity to identify the walking user rapidly. So, we combined the walking features with touch features to establish the SVM model for dynamic scenario. To evaluate the identification performance of *SilentSense* in dynamic scenario, 50 volunteers are required to use phones while they are waking freely. Our experiments show that the FAR and FRR reduce to 0 after only about 3 steps. Considering the different amount of training data, after 12 steps, the accuracy to identify a guest can achieve 100%, and after 7 steps, the accuracy to identify the owner can achieve 100%.

3.8 Summary

In this chapter, a framework is designed to verify whether the current user is the legitimate owner/user of the smart device based on the behavioral biometrics, including touch behaviors and walking patterns so as to protect private information stored locally. We establish a model and a novel method to silently verify the user with high confidence: the

false acceptance rate (FAR) and false rejection rate (FRR) could be as low as $< 0.1\%$ after only collecting about 3 observed actions. One noticeable discovery is that a user's touch signatures if used in conjunction with the walking patterns will achieve significant low error rates for user identification in a completely non-intrusive and privacy preserving fashion. Previous works have shown the feasibility of using user touch behaviors for authentication, but they either are tailored for enhancing the passcode typing security, or only work for a special application. I believe that our approach can also be used to distinguish users for multi-user touch display and gaming.

CHAPTER 4: PRIVACY CONCERN EXPRESSION AND PROTECTION: SHARING

4.1 Photo Privacy: Practices and Challenges

The imagery privacy policies are designed as extension of policies of protecting 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-enabled 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 [9]. Similarly, a bill of "Camera Phone Predator Alert Act" [2] was also proposed in 2009 in US for the same purpose.

Consent:

In addition to disclosing their practices, some organizations should 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 [71]. For photo sharing, the current social norm is that the subjects are opted in by default.

Damage Control:

Personal data can be compromised through social media and cause unexpected damage [81,92]. When private information is leaked, some online service providers allow users to contest and 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 since it involves two parties: (1) the photographer (broadly defined to refer to anyone who take pictures) and (2) 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, for instance, 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 photo-takers to deliberately avoid taking photo of them [16]. Some online service providers voluntarily take steps to protect imagery privacy. For example, both Bing StreetSide and Google Street View [5] blur all people's faces and vehicles' license plates [6] in the images before serving them publicly. However such a blurring-all solution is obviously not ideal for photos sharing scenarios.

In this case, we propose PRSP, a Privacy Respecting and Respecting Protocol, which consists of a Privacy.Tag and a Privacy Respecting Sharing Protocol (PRSP). In our protocol, the Privacy.Tag enables a user to explicitly and flexibly express their privacy deal,

while the PRSP empowers the photo service provider to exert privacy protection following users' policy expression so as to mitigate the public's privacy concern.

4.2 Challenges

Since 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.

4.3 Concept and Design Overview

The above challenges will be addressed through the protocol deigns, associated with concept of he protocol as well as the tag design.

4.3.1 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) [86] 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.

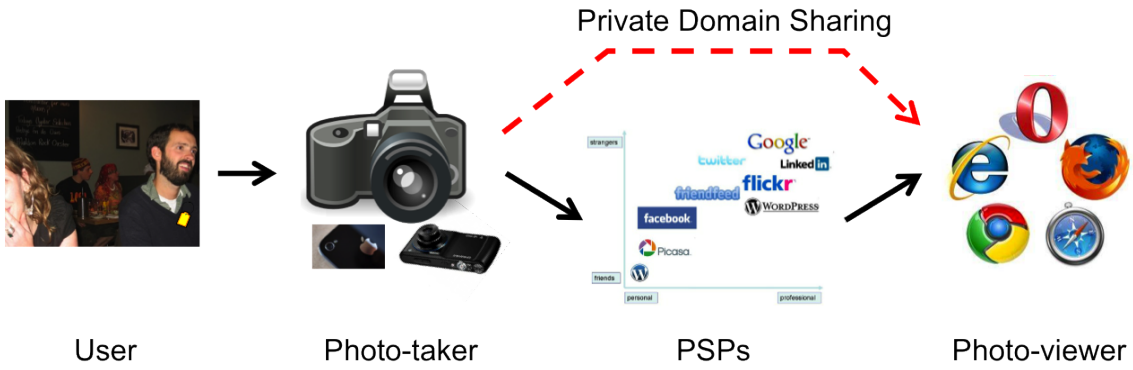
4.3.2 Design Considerations

(1) Diverse Privacy Appeals

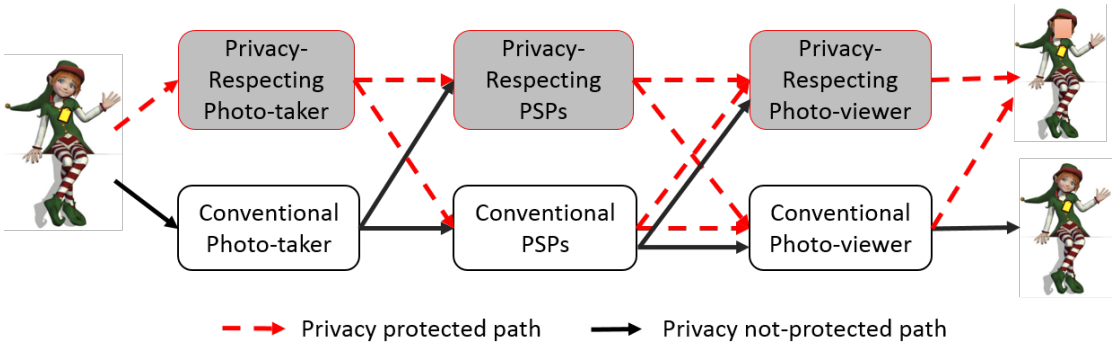
In real life, different users have different privacy appeals; even for the same person, the privacy desire may be situation dependent. We empirically classify privacy appeals regarding photo-sharing into three general categories:

- 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



(a) Photo/Privacy propagation chain



(b) Impact of privacy protection at different stages

Figure 20: Photo and privacy propagation chain and impact of privacy protection at different stages.

happens at social gatherings.

- 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.

(2) PSPs Being the Narrow Waist:

Figure 20-(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 for multiple reasons. 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 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 difficult to say that the photo sharing policy of the subject is violated. Therefore, we propose that as long as a photo is not shared out, there is no privacy 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. And it would be the most effective way to protect people's privacy. However, given the large amount of cameras user already own and the wide penetration and huge diversity of new-camera equipped mobile devices, deliberately posing any assumption on the camera is not a good option. Therefore, the solution of requiring the cooperation of PSPs and users' privacy desire is another assurance.

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 [89], 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.

4.3.3 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
```

Generally, 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 clearly. 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 [30,49]. 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 `url` with all user's privacy requirements into the QR-code alone may simplify

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

the QR-code design, we still encourage the user to specify partial of the policy in the tag.It is due to the fact that 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. In real scenario, 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.

4.3.4 Privacy Expression and Respected Protocol Design

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

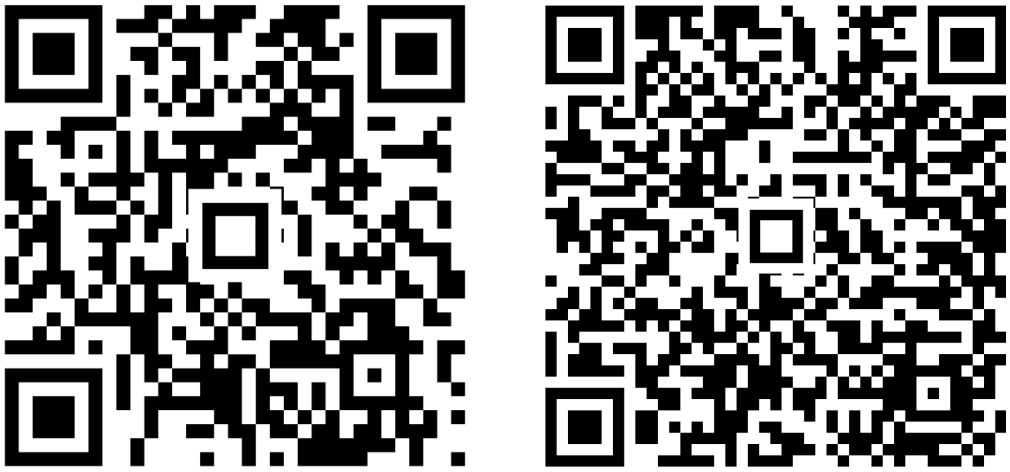
(1) On The User Side:

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.

(2) 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.*



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

Figure 21: Proposed QR-code based Privacy.Tag design, with a color-reversed position locator at the center

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.

(3) 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.

4.4 Privacy.Tag Realization

In this section, I will describe and justify my design of a practical Privacy.Tag.

4.4.1 Required and Desired Tag Properties

(1) 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.

(2) 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.

4.4.2 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, but with following additional features.

(1) Customized Yet Compatible 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 21. 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 a normal QR-code, as long as the QR-code can be detected, no matter it is successfully decodable or not.

(2) 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.

(3) 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 [11], 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 [12]: 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. I will present more detailed study on the impact of tag sizes and justify our decision in Section 4.7.1.

4.5 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 22. 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

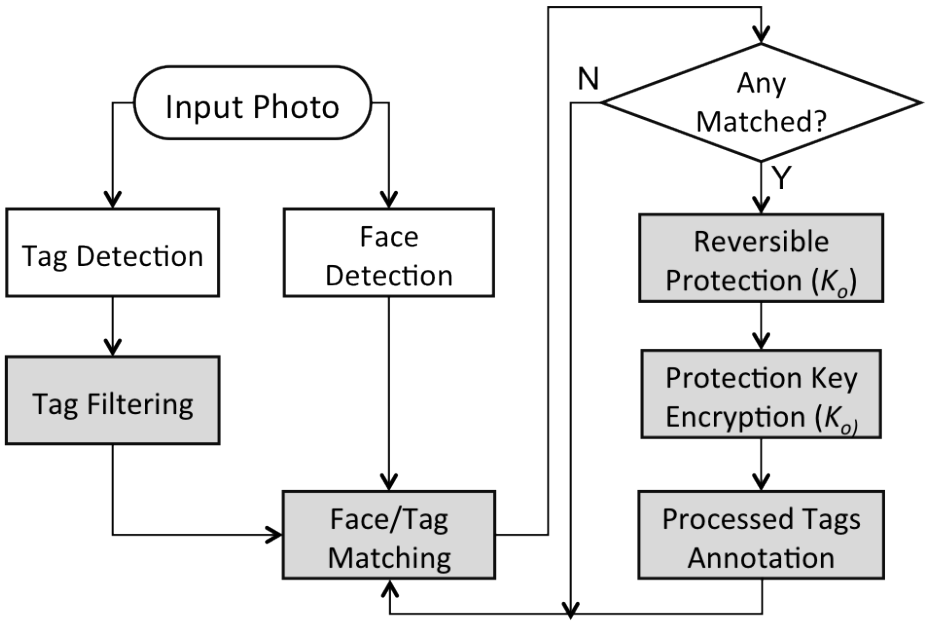


Figure 22: General privacy protection procedure.

photo privacy protection, users want to prevent their face from being recognized.

4.5.1 Face/Tag Matching

When a photo is shared, the privacy protector (i.e., PSPs) will perform face detection and also Privacy.Tag detection. However, in many cases, there are 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 process 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.

Here, we propose a *Range-constrained face/tag matching* process. Essentially, the size of the face is easily obtained from the face detection module, say H long and W wide, and

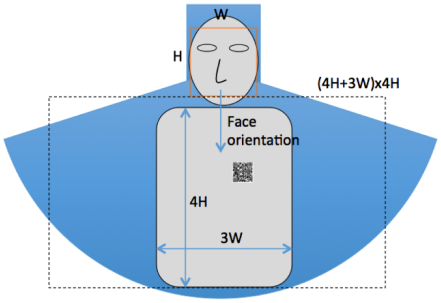


Figure 23: Search area illustration in face/tag matching

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 23. 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. 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.

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

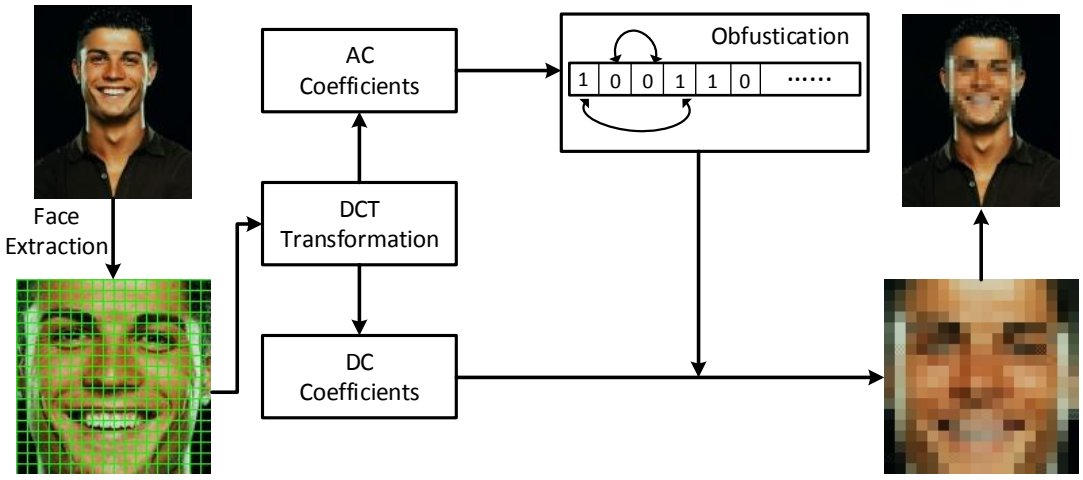


Figure 24: Procedure of proposed secret block-based obfuscation process for privacy protection.

future work.

4.5.2 Reversible Protection

Commonly adopted privacy protection strategies include face blurring or mosaicing [5, 42]. 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.

(1) 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 24. 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 inter-dependent.

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 24 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.

(2) 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.

4.5.3 Obfuscation Key Encryption

On the other hand, we also wish 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 stor-

age 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.

4.5.4 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.

4.6 Implementation and Evaluation

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.

4.6.1 Key Components

(1) Face detection:

Improve the performance of face detection is out of the scope of this paper. We simply

adopted a Microsoft FaceSDK [11], 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.

(2) 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 4.5.1.

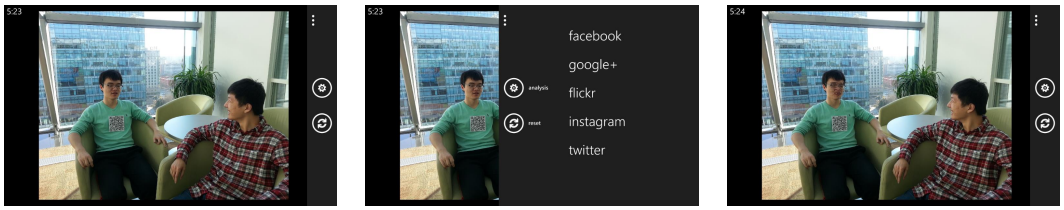
(3) Tag annotation embedding:

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 [101]. 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.

(4) 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.



(a) Selecting photo to share (b) Selecting target PSP (c) Upon sharing

Figure 25: Privacy.Tag Implementation in Windows Phone 8.

4.6.2 Prototype

We have fully implemented the proposed PERP on Windows Phone 8 platform. Figure 25 shows a real use case. The user was about to share a photo of two people in a casual chatting to Google+. The person wears a Privacy.Tag say only Facebook is allowed to show unprotected face. When the photo is being shared to Google+, the face is protected. Thus, upon sharing, the person's face are protected.

4.7 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.

4.7.1 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.

(1) 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 26 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

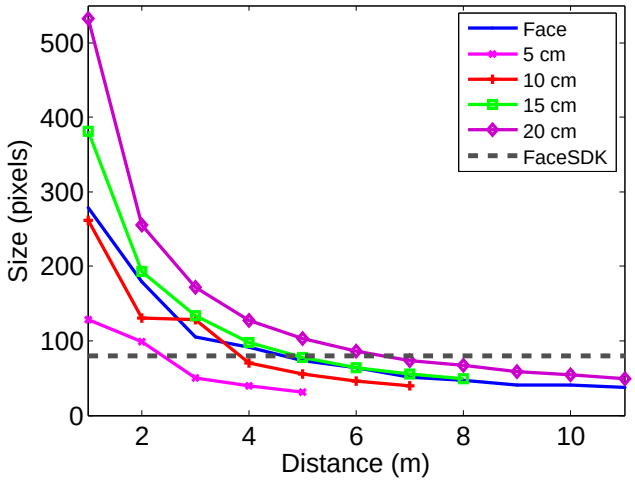


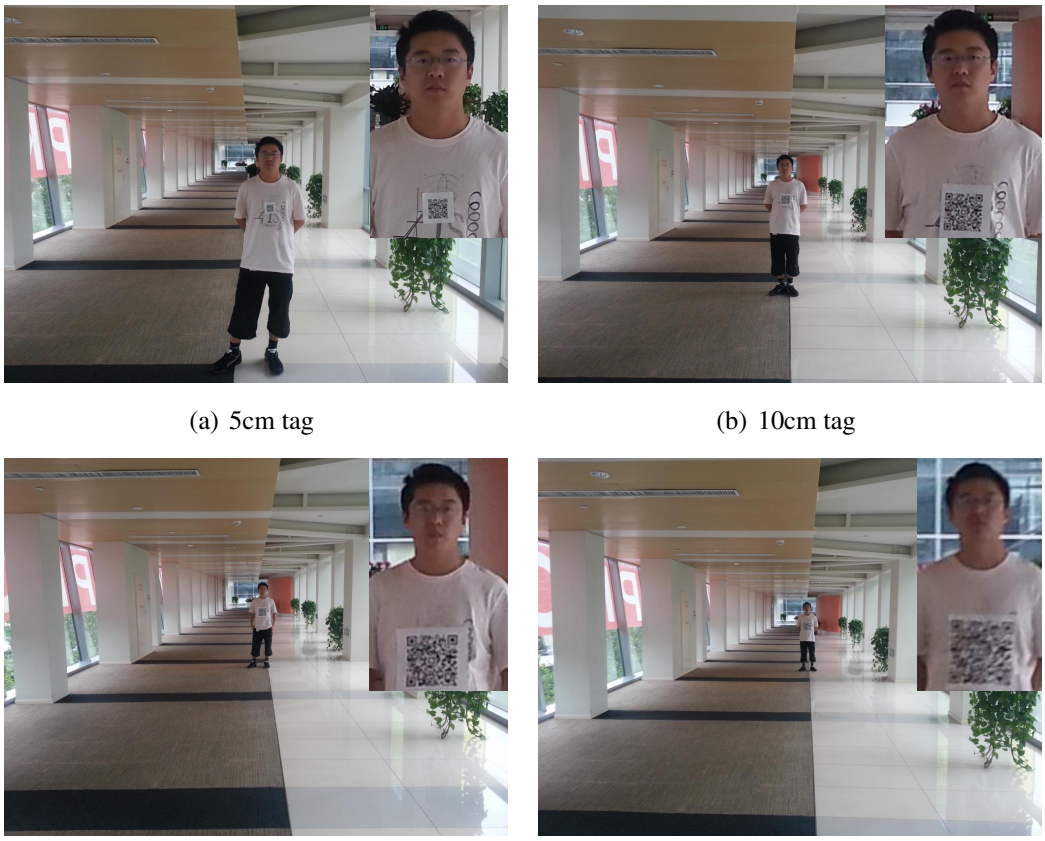
Figure 26: Size of face and tags in the photos taken at different distances.

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.

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.

[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

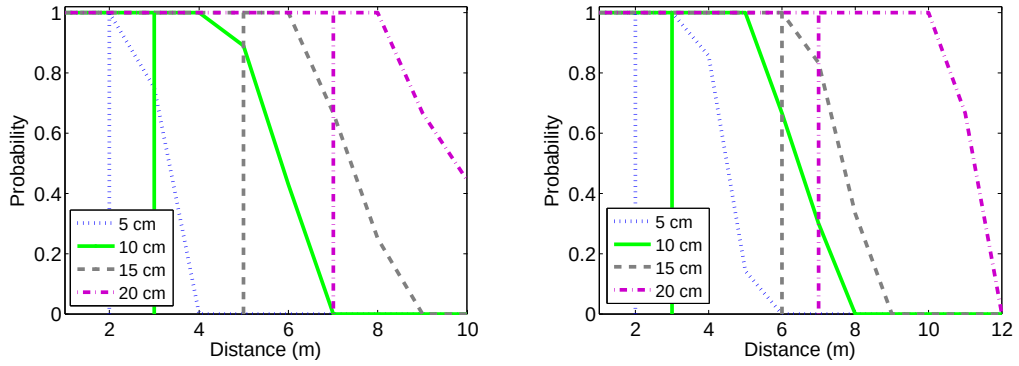
(c) 15cm tag (d) 20cm tag

Figure 27: Sample pictures showing different sized Tags at their maximum detectable distances.

To illustrate the real situations, Figure 27 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.

(2) 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 28



(a) Indoor (office) (b) Outdoor (park)

Figure 28: Detection probability different sized tags (carrying 32Bytes) in indoor and outdoor environments.

shows the detection probability for different Tag sizes (all carrying 32 Bytes information, as shown in Figure 21(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 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.

(3) 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.

Table 2: Reliable decoding distances vs amount of embedded information, for 5cm and 10cm tags.

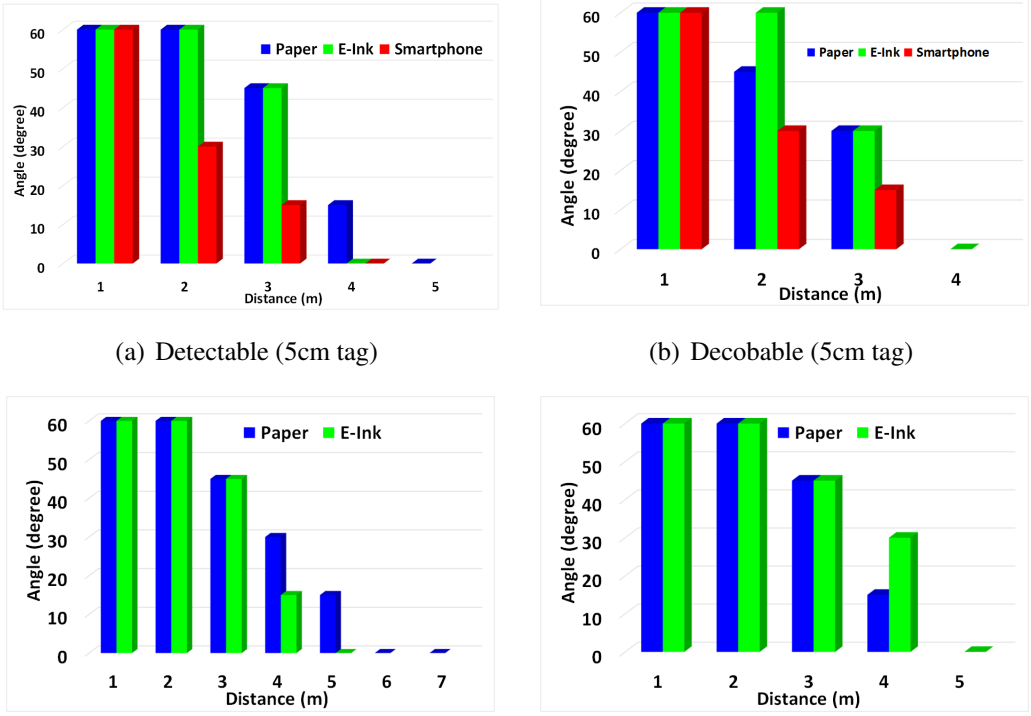
	Materials	16B	32B	64B	128B
5cm	Paper Tag	3m	2m	1m	1m
	E-Ink	4m	3m	2m	1m
	Smartphone	3m	2m	1m	1m
10cm	Paper Tag	4m	3m	2m	2m
	E-Ink	5m	4m	3m	2m

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 2 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.

(4) Impact of Shooting Angles:



(a) Detectable (5cm tag)

(b) Decobable (5cm tag)

(c) Detectable (10cm tag)

(d) Decobable (10cm tag)

Figure 29: Tag detecting and decoding at different angles across different shooting distances.

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 29. All three kinds of Tags could

be reliably decoded when the shooting distance is one meter, 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.

4.7.2 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.

(1) 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.

(2) 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 3: Face/Tag Matching in Real Situations

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

(3) Results:

The precision and recall for both indoor and outdoor scenarios are presented in Table 3. 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 occlusions 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.

4.7.3 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 4: 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

Table 4 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 [11]. 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 [88]. 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.

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.

4.8 Discussions

The exemplar design of QR-code based Privacy.Tag design and evaluation various aspects of it has been presented in the previous sections. In 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 my 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. My 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. I do not attempt to claim that it is the only viable solution - there still exist plenty of scope for improvement. Similarly, I have designed a reversible protection scheme through a secret obfuscation process. Many alternative or better ways can be designed.

The adopted standard QR-code has a very limited capacity, which has in return severely constrained the amount of information we can embed into the tags and the decodable range. The 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. Essentially, efforts could be put to beautify the current QR code by introducing stylish design [31] without affecting the performance, such as Halftone QR Codes [36], Visualead QR codes [17]. Or just using adopt new kind of picture-embedding 2D barcode, such as PiCode [57].

I 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.

Although a QR-code is adopted as a concrete embodiment of a Privacy.Tag, the solution is not limited to QR-codes. Although not everyone is willing to wear QR-codes all around their outwears, I 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. I 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. In the future, 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. I 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.

4.9 Summary

In this chapter, I 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. The PERP has been fully implemented and various aspects of the Tag design have been evaluated, the protection performance and the computational overhead of the Protocol. The results confirm the technical feasibility of PERP. Therefore, advocate PSPs to collectively follow the Protocol and contribute to a healthy photo sharing ecosystem.

CHAPTER 5: PROTECTING IMAGE AND VIDEO IN PUBLIC: BROADCASTING

5.1 Preliminary

When it comes to protecting the content of both images and videos displayed in public, one feasible solution is to design a special video type, a *watch-only video*, *i.e.* the video can be displayed on common devices and be watched by human with the same visual quality as the original video, but the pirate version, captured by pirates' mobile cameras, will suffer a severe quality degradation. This is challenging as nowadays mobile devices are equipped with sophisticated cameras which are imitation of the human eye. Before presenting the design, I would briefly review the cutting-edge display-camera communication as preliminary, followed by the properties of human eye as the information receiver and the constraints that the display and camera technologies place on the transmission of the light signal.

5.1.1 Display-Camera Communication

Display-camera communication has been attracted most researchers' attention in the last few years, which employ one-way video stream to transmit information [54]. PixNet [82] firstly leverages 2D OFDM to modulate high-throughput 2D barcode frame, and optimizes its capacity in screen camera communication channel. COBRA [52] then achieves real-time phone-to-phone optical streaming by implementing a newly designed special code layer, which support fast corner detection as well as blur-resilience technology. VRCodes [111] propose a new unobtrusive barcode design, which is imperceptible to human eyes. LightSync [55] adopts rolling shutter effect to address the imperfect frame synchronization.

Strata [56] supports various frame capture resolutions and frame rates to deliver corresponding information correctly. Besides, Hilight [70] transmits the information by dynamically adjusting the hues of the image, and InFrame [106] achieves dual-mode full frame communication between screen and both humans and devices simultaneously.

5.1.2 Characterizing Human Vision

Human possess a photopic vision system, which is driven by the cone-cells in the retina. When we see the rich light spectra of objects, different light wavelengths stimulate the three kinds of cone-cells of a viewer in different degrees, providing her perception of distinct colors. Color is usually recognized by the viewer with two aspects: (1) *luminance*, which is the indication of the ”brightness” of the light; (2) *chromaticity*, which is the property that distinguishes the composition of the light spectra.

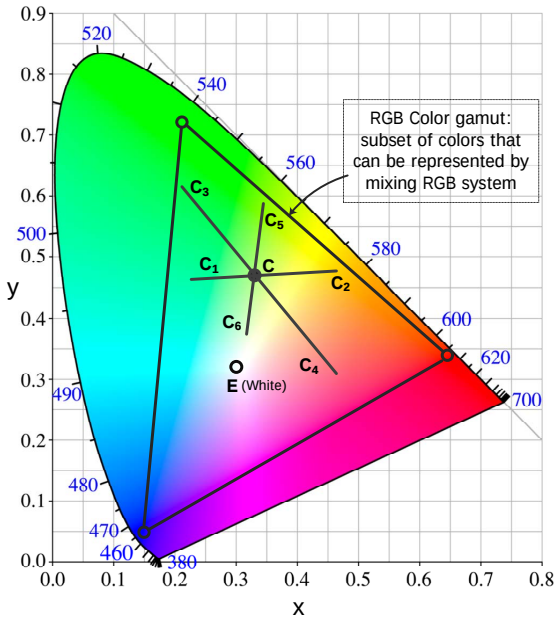


Figure 30: CIE 1931 chromatic diagram and color mixture.

Various models are designed to quantify human color vision. The commonly used 1931 CIE color spaces are the first defined quantitative links between the physical pure colors

(*i.e.*, wavelengths) in the electromagnetic visible spectrum and the physiological perceived colors in human color vision. It converts the spectral power distribution of light into the three tristimulus values X, Y, Z. Here Y determines the illuminance (brightness), and X and Z give chromaticity (hue) at that luminance. The chromaticity values can be presented in a CIE chromatic diagram as illustrated in Figure 30, where coordinates are defined by $x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$. The diagram represents all of the colors visible to the average person. In the rest of this chapter, I will use (x, y, Y) values to describe the chromaticity and illuminance of a specific color.

The colors along any line-segment between two points can be made by mixing the colors at the end points, which is called the *chromatic additive rule*. Suppose we have two colored light C_1 and C_2 with values (x_1, y_1, Y_1) and (x_2, y_2, Y_2), and mix the two colors by shining them *simultaneously*, we obtain the mixed color (x, y, Y) denoted by:

$$\begin{cases} (x, y) = \frac{Y_1}{Y_1+Y_2}(x_1, y_1) + \frac{Y_2}{Y_1+Y_2}(x_2, y_2) \\ Y = (Y_1 + Y_2)/2 \end{cases} \tag{4}$$

The rule shows that, the chromaticity for the mixed color lies on the line segment joining the individual chromaticities, with the node position on the line segment depending on the *relative brightness* of the two colors being mixed. Clearly, the combination of colors to produce a given perceived color is not unique. For example, the pair C_1C_2, C_3C_4, C_5C_6 in Figure 30 can each produce the same color C if combined in the right proportions.

When people watch temporal varying colors, they receive both illuminance change and chromaticity change. When two isoluminant colors alternate at frequencies of $25Hz$ or higher, an observer typically perceives only one fused color, whose chromaticity is deter-

mined based on the chromatic additive rule previously discussed. This may also relate to *persistence of vision*, the theory where an afterimage is thought to persist for approximately $\frac{1}{16}$ of a second on the retina, which is also believed to be the explanation for motion perception. Figure 31 illustrates the color fusion result by human eyes. For example, people perceive alternate red and blue as magenta, and alternate red and green as yellow.

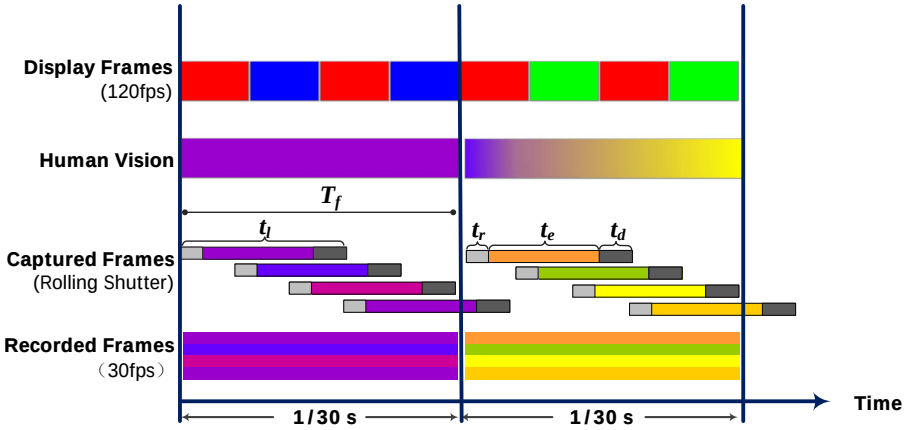


Figure 31: Color perception by human eyes and image capturing by CMOS cameras.

Although, human's visual system is very efficient and powerful, its ability to interpret the temporal information presented on video displays is limited. When the change frequency is smaller than eye's *temporal resolution*, called *critical flicker frequency* (CFF), *flicker* happens [24, 59, 76]. Illuminance flicker is a visible illuminance fading between frames displayed on screen when the brightness drop for time intervals sufficiently long to be noticed by a human eye. Chromatic flicker is defined similarly. On the other hand, when the flicker frequency is larger than the CFF threshold, the illuminance flicker stimulus and chromatic flicker stimulus from a sequence of continuous frames are only perceived by human as time-averaged luminance and time-averaged wavelength respectively.

Typically, human eyes can only resolve up to $50Hz$ to *luminance flicker* and $25Hz$ to

chromatic flicker [59]. Thus our eyes cannot capture fast moving objects and high frequency flickery images [106]. In practice CFF depends on both the spatial and temporal modulation of luminance across the display [44,103]. If the absolute amplitude of the main frequency of the display luminance modulation is greater than a pre-determined frequency-dependent threshold (denoted as $A(f)$) the observers will perceive flicker. Typically

$$A(f) = a \cdot e^{bf} \tag{5}$$

where f is the refresh frequency and a and b are constants which depend on the size of the luminous area.

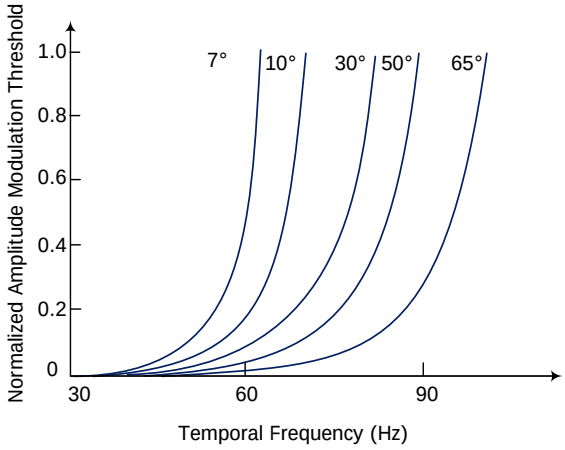


Figure 32: Flicker regression equation for different display field sizes.

Figure.32 illustrates the equation for different display field sizes where the degree value measures the angle of the smallest cone apexed at eye to cover the display field. Then we have $CFF = \frac{\ln[A(f)/a]}{b}$.

5.1.3 Video Encoding and Display

Most screens produce a wide range of colors by stimulating the cones of eyes with varying amounts of three primary colors - red, green and blue. The most widely used display

devices are LCD monitors and projectors. We cannot display the full range of human color perception with these devices, because the gamut of normal human vision covers the entire CIE diagram while the gamut of an RGB display can be represented as a triangular region within the CIE diagram, with three vertexes are red, green and blue (see Figure 30).

Video is typically stored in compressed form to reduce the file size, and a number of video file formats were developed. In this work, we will consider a generic video stream consisting of sequential still images, referred to as *frames*. Each frame is a matrix of color pixels. During playing the video, the video stream is decoded and presented by the display system frame by frame. Displaying frames in high frequency (or called *refresh rate*) creates the illusion of moving images because of the phi phenomenon [48]. Modern off-the-shelf LCD monitors and projectors support $120Hz$ refresh rate, and the refresh rate for some game LCD monitors could reach $240Hz$. Since most films are shot in 24 or 30 frame per second, it is common that each original movie frame will be repeated several times while being displayed on screen. In proposed system, instead of repeating each original video frame directly, I will carefully design these *displayed frames* by changing the color pixel such that the viewing experience of live audiences is not affected and it also prevents high-quality videotaping from third-party cameras.

5.1.4 Video Recording

When a video is displayed in screen, two communication channels will be investigated: *screen-eye* channel and *screen-camera* channel. The screen-eye channel represents how a human will perceive the displayed video, which has been discussed in Subsection 5.1.2. Here I will briefly review some important specifics of screen-camera channel, which later will be used to design the watch-only video.

The camera sense the color quite similarly as human eyes. Each pixel receives light of different wavelength during the exposure time, and fuses them to compute the illuminance and chromaticity values of this pixel. Onboard cameras now could capture high-resolution mega-pixel images at fast frame rate (called *record rate*), which even exceed the perception capability of retina. For example, the record rate of traditional onboard cameras is 24, 30 or $60fps$, while some of latest mobile smartphones, *e.g.*, iPhone 5, iPhone 6 and Samsung Note 4, support up to 120 and $240fps$ in high quality.

Currently, CMOS image sensors have become mainstream in onboard cameras for mobile devices, which expose and read-out each rows of pixels consecutively [87]. Most of consumer-grade cameras implement such image sensor due to its low energy cost, but this leads to geometric distortion of captured image, called *rolling-shutter effect*. We explain the rolling-shutter mechanism by a simple example as shown in Figure 31. Assume that, before exposure, each line of a video frame requires duration of t_r second by the camera sensor for resetting the line to query the data. The sensor scans the scene line by line to synthesize the complete image, and for each line, the duration for the sensor exposed to the light is t_e before it takes t_d line scan acquisition time for the driver to dump the data. Then the total acquisition duration (denoted as t_l) for retrieving a line is

$$t_l = t_r + t_e + t_d. \tag{6}$$

Assume the recording rate of a camera is $30fps$, so that the duration for constructing a single frame is $1/30s$, denoted as T_c. Since each frame contains multiple batches of line scans with each line of duration t_l, which are exposed and dumped sequentially and overlapped in parallel, we define *effective light sampling frequency* as the number of lines being captured

in one second. Although typical rolling-shutter camera captures an image at its reported frame rate $f_c = \frac{1}{T_c}$, its effective sampling frequency is $f_s = f_c \times n$, where n denotes the actual number of lines in individual images.

Generally, the shutter is required to open for a certain duration for sufficient light to complete a single frame, and cameras generate single frame continuously in pre-defined high frequency to record a video. The exposure duration depends on the sensitivity of sensor itself and the actual lighting condition, including the contrast and intensity. Most consumer-graded cameras adjust their frame rate automatically to ensure the frame visual quality for the whole captured video. According to the experiments conducted by [55], some off-the-shelf mobile devices cannot reach nominal frame rate when recording, and the inter-frame time intervals often fluctuate.

5.2 System Design Overview

Therefore, I propose KALEIDO, a light-weight hardware-free secure video encoding and displaying system. In this section, I will discuss the design space and principles of KALEIDO, the design challenges and opportunities in detail.

5.2.1 Design Space And Principles

Suppose an original video is produced with 24fps or 30fps movie rate, and the video will be displayed in some screens with larger refresh rate, *e.g.* $120fps$. The goal is to protect the copyrighted video from undesired recording by commercial mobile devices with diversified recording rates, instead of prohibiting the display of the video using mobile devices. The pirate shooting is limited to those using commercial onboard cameras of mobile devices. The professional high-end film cameras are thus excluded. We aim at designing a more radical and effective method to generate a legitimate watch-only version of the videos from

the original video. For each original movie frame, we will generate a sequence of watch-only frames (precisely $\frac{\text{refresh rate}}{\text{movie rate}}$ frames). The watch-only video can be displayed by any off-the-shelf display device. When the watch-only video is displayed normally, viewers will not notice any quality difference from the original one, *e.g.*, without color distortion, artifacts or flicker. But when watching the pirated version recorded by a camera, viewers will suffer a severe intolerable quality degradation.

Leveraging opportunities offered by the limited resolution of human vision system, rolling shutter of the camera and asynchronization between display and camera systems, we propose a system proposed system which takes the original video as input and produces a *watch-only* version. As shown in Figure 33, proposed system is an add-on for the current video play system without extra hardware. The piracy procedure typically consists of four steps:

1. the legitimate video is re-encoded and displayed;
2. the pirate shoots the displayed video with a camera;
3. the captured frames are recorded into a video file, which is the pirate version of the original one;
4. the pirate video is displayed for the viewer.

The key of the solution is to re-encode the original video into a *watch-only* one, under the constraint that the viewer's watching experience should be reserved in the first step; but after the second and third steps, the watching experience degradation should be maximized at last.

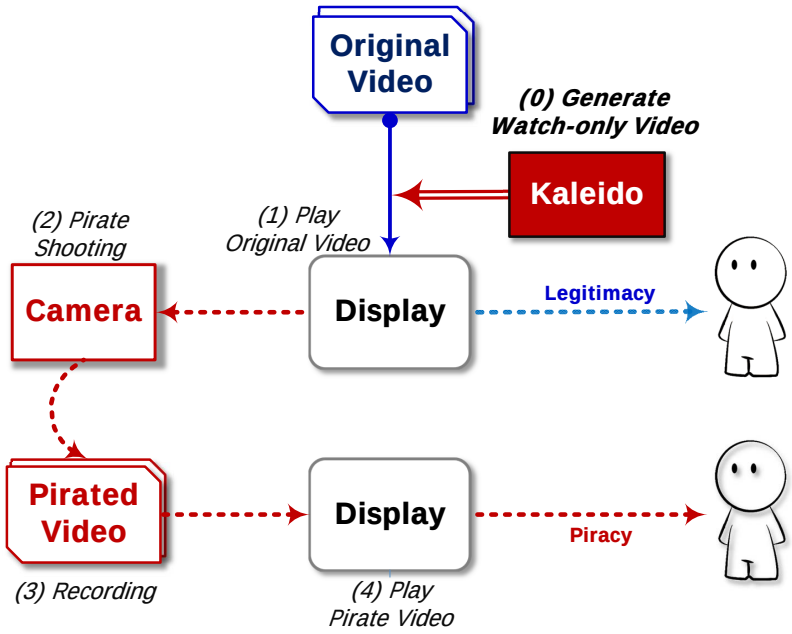


Figure 33: Original display v.s. pirated video display.

5.2.2 Design Opportunities

One basic approaches for causing a quality loss into the viewing experience of the pirated video are to introduce illuminance flicker and chromatic distortion into the re-encoded frames. The most challenging part of proposed system is to ensure the encoded flicker and distortion are imperceivable to the legitimate viewers at first, and then become perceivable after a piracy procedure.

To address these challenges, I will reinvestigate the disparity between the human vision system and the camera system. As shown in Figure 31, the human eye receives light illuminance and chromatic perturbations in a continuous but low-pass manner, while camera captures light as a discontinuous sampling system with a higher temporal resolution. Taking the continuous frame stream as a varying light signal with specific spatial and temporal color distribution, we exploit the information loss and distortion by camera shooting to

look for opportunities. Let the refresh rate of display be f_d and frame record rate of camera be f_c. Then the display duration for each frame is $T_d = \frac{1}{f_d}$ and recording window of a recorded frame is $T_c = 1/f_c$. I will analyze the following two complementary cases.

Case 1: $f_d > f_c$. (Display rate is larger than record rate)

In this case, there are multiple frames displayed during a single capture time window T_c by the camera. Remember that the rolling shutter effect of camera causes a line's exposure time t_e be less than T_c. In practice t_e could be less than the half of T_c. Hence, for a specific line in the recorded frame, its exposure time is not enough to record the complete light signal during T_c. If the signal is time-invariant, the line doesn't lose any information. That's why I can record a traditional video ($30fps$) displayed on a $120Hz$ screen using a $30fps$ camera, since it repeats each frame four times. If the signal is time-varying (*i.e.*, re-encoded frames from a single original frame are different), then part of the variation cannot be recorded, *i.e.*, the temporal distribution of the recorded signal in the pirate video deviates from the original video frame. Because eye perceives time-averaging chromaticity and illuminance, the temporal variation loss could cause a perceivable distortion. Besides, different lines in the recorded pirate video lose different portions of the temporal variation, which could cause a spatial deformation of each recorded frame. Figure 31 presents an example, where $f_d = 120Hz$ and $f_c = 30fps$. It is a common setting for commercial display devices and cameras. In this figure, two intervals defined by three vertical lines represent two original video frames. The first row represents four encoded display frames for each of the original video frame. During one capture time window, four frames display alternate colors in this case. The second row denotes the colors to be perceived by human eye. The human vision perceives one color by fusing them equally. The third row denotes

the video capture procedure by camera with rolling shutter effect, while the fourth row denotes the recorded two frames by the camera. Each line of the camera captures only part of the display frames, which results in distorted color fusion results for each line. And the recorded image presents a stripped pattern.

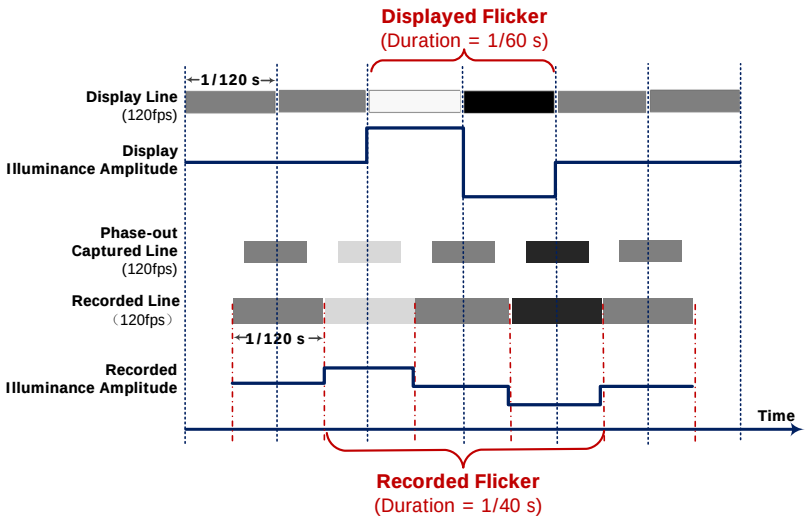


Figure 34: Flicker pollution due to out-phase camera sampling.

Case 2: $f_d \leq f_c$. (Display rate is less than record rate)

In this case, every displayed frame can be captured by at least one recorded frame. If the display system and camera system are ideally synchronized, then during the exposure time of any line of the recorded frame the light signal is constant (from a single display frame) and the camera can record the displayed light signal with high fidelity. In practical applications, with high probability, the camera is asynchronized with the display, which cause out-phase lines in each recorded frame. As illustrated in Figure 34, one out-phase line captures light signal from two successive displayed frames. If there is a flicker (two successive darker and lighter frames, or vice versa) at a frequency $f_f = f_d/2$, the perturbation will be captured by $2\frac{f_c}{f_d} + 1$ temporal successive out-phase lines. As a result, the flicker

is recorded but its frequency is down-converted to $\frac{2}{2+f_d/f_c} f_f$. In the example of Figure 34, the flicker frequency is down-converted from 60Hz to 40Hz. With this observation, we have an opportunity to encode invisible noises, whose frequency is larger than CFF, to the original video. After the down-conversion by camera recording, the noise could become visible because its frequency now falls below the CFF.

Additionally, unstable inter-frame intervals of most commercial onboard cameras aggravate the information loss and distortion for both Case 1 and Case 2, hence making the color distortion of recorded frames even worse.

Consequently, during a single capture time window (T_c), rolling shutter effect and unstable intervals could cause a temporal information loss or deviation in the recorded frame if the displayed frames for each original video frame are time-varying. We then propose two techniques to aggravate the temporal information distortion.

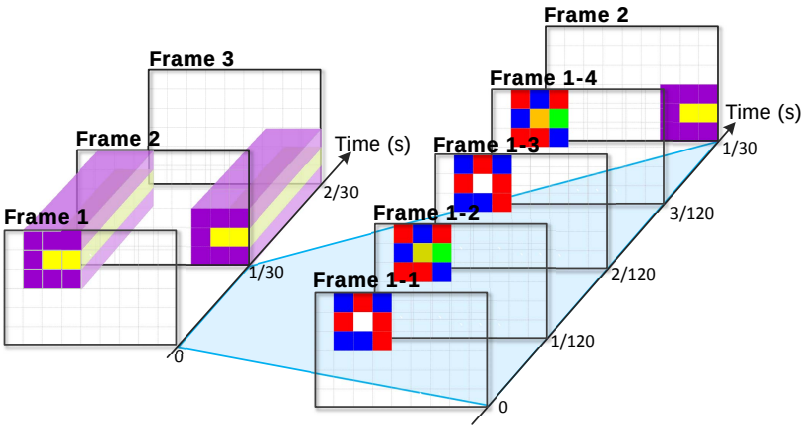


Figure 35: Color decomposition for display frames.

Technique 1: Chromatic Frame Decomposition.

Since most current videos are $24fps$ or $30fps$, with high refresh rate display devices (*e.g.*, $120Hz$) one video frame can be decomposed into n (*e.g.*, 4 or 5) successive dis-

play frames following the temporal chromatic additive rule of human eyes. We propose to decompose one invariant chromatic signal into chromatic flickers, which can be fused by human eye. Figure 35 shows an example of the color frame decomposition. The chromatic flicker frequency is 60, which is larger than the chromatic CFF. Note that our design is different from the visual cryptography [78], based on the visual effect produced by overlapping multiple transparent slides.

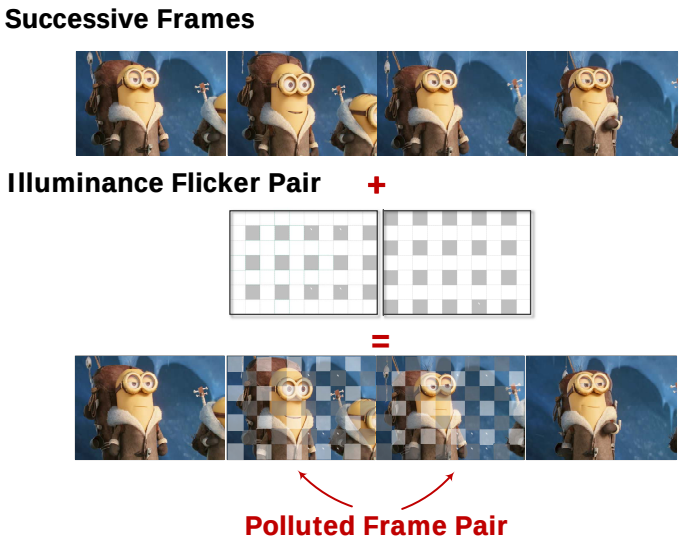


Figure 36: Illuminance frame pollution in display frames.

Technique 2: Illuminance Frame Pollution.

When there is illuminance fluctuation, human eye works as a low pass filter to eliminate the high-frequency flicker and perceives the averaging illuminance. We propose to add imperceivable illuminance flickers to pollute the frames. As illustrated in Figure 36, each flicker is a pair of pollution frames. The time averaging illuminance of each pixel from two pollution frames equals 0, which cancels out illuminance change for human eye. But if there is a temporal information distortion in the recorded frames, the flicker cannot be

balanced out. Besides, the flicker's frequency is just above the illuminance CFF and its amplitude will be maximized. So if any down-conversion happens, the flicker will become perceivable.

Technique 3: Embrace Spatial Deformation.

In company with temporal distortion achieved by chromatic frame decomposition and illuminance pollution, we also design proposed system to deform each decomposed frame's shape to prevent image capturing during the video play. Our goal is to make display frames' colors appear as random as possible. Randomizing each display frame's color, while preserving the view experience of legitimate audiences, is possible due to the metamerism. Note that a color can be decomposed to an infinite number of different color pairs. Randomizing different decomposition color pairs will make each display frame like a random noise.

5.3 Watch-Only Video Generation

We are now ready to present KALEIDO by exploiting the main techniques (chromatic frame decomposition, illuminance frame pollution, and spatial deformation) and integrating them to generate watch-only videos.

For simplicity, we consider a $30fps$ video in our current system design, which consists of N sequential frames with $\{V^1, V^2, \cdots, V^N\}$. Each frame V^k is a $R \times C$ matrix, with each pixel's P_{ij}^k color is $C_{ij}^k = (x_{ij}^k, y_{ij}^k, Y_{ij}^k)$. Recall that, (x, y) determines the pixel's chromaticity, *i.e.* coordinates in the CIE diagram, and Y is the illuminance level. In KALEIDO, we focus on an off-the-shelf display device. As a running example to demonstrate our design, we assume that the refresh rate of the display device is $120Hz$. Our scheme can easily be modified to adapt to different refresh rates. We decompose each original frame V^k into 4

display frames (called *sub-frames*) $\{V^{k,1}, V^{k,2}, V^{k,3}, V^{k,4}\}$. Note that all sub-frames have the same duration $1/120s$. To guarantee the flicker frequency greater than CFF and the sub-frames can be fused by human eye, we decompose each frame into two different sub-frames, referred to as fusion pair, and repeat the fusion pair. We then need to determine the $(x_{ij}^{k,l}, y_{ij}^{k,l}, Y_{ij}^{k,l})$ values of each pixel $P_{ij}^{k,l}$ in sub-frames. For $24fps$ video, it can be easily converted to $30fps$ using standard pulldown tools, or each frame can be decomposed to 5 frames, which makes the decomposition more complex, but the principle and techniques are the same.

According to the chromatic additive rule and flicker fusion rule, given the color of a pixel $C = (x, y, Y)$, we need to decompose it to two colors $C_1 = (x_1, y_1, Y_1)$ and $C_2 = (x_2, y_2, Y_2)$, which satisfies

$$\begin{cases} (x, y) & = \alpha(x_1, y_1) + (1 - \alpha)(x_2, y_2) \\ Y & = (Y_1 + Y_2)/2, \end{cases} \tag{7}$$

where $\alpha = \frac{Y_1}{Y_1+Y_2}$. Since the mixed chromaticity (x, y) is a weighted average depending on the relative illuminance of the decomposed two colors, we should determine the illuminance of each pixel first. So, based on the pixel illuminance of the original video, KALEIDO firstly determines the illuminance pollution of the sub-frame sequence, which gives the final illuminance level (*i.e.*, Y values) of every pixel. Then the illuminance ratio α is fixed. Secondly, (x_1, y_2) and (x_2, y_2) are selected to (approximately) maximize the temporal distortion and spatial deformation.

5.3.1 Illuminance Frame Pollution

Let the initial illuminance levels of every pixel in sub-frames equal to the illuminance in its original frame, say Y_{ij}^{k}. Given two successive sub-frames $\{V^{k,1}, V^{k,2}\}$, a pixel pair $P_{ij}^{k,1}$ and $P_{ij}^{k,2}$ have illuminance levels $Y_{ij}^{k,1} = Y_{ij}^{k,2} = Y_{ij}^{k}$. If we can add an illuminance complementary perturbation $(+\delta, -\delta)$ to the pixel pair, it changes their illuminance levels to $Y_{ij}^{k,1} + \delta$ and $Y_{ij}^{k,2} - \delta$. Then the human eye perceives an average illuminance Y_{ij}^{k}, which equals the original illuminance level if the refresh frequency is above the CFF. In this way, the added complementary perturbation is imperceivable. However, when there is a temporal information loss (as in Case 1), the perturbation cannot be canceled out; when out-phase captures happen (as in Case 2), the perturbation's frequency is down-converted. In those situations, the perturbation becomes perceivable flicker to human.

Based on this rule, we can add imperceivable flicker (either $(+\delta, -\delta)$ or $(-\delta, +\delta)$) to pixel blocks of two successive sub-frames to pollute the displayed video. The values of the amplitude δ and block size should be maximized to aggravate the pollution. Remember that the CFF increases as greater amplitude and larger block size, which could cause the flicker perceivable once CFF $> 60Hz$. For example, the CFF is about $90Hz$ when the block size, together with a viewing distance, resulting a 65° angle of view and the normalized amplitude is greater than 0.4 (Figure 32). So our pollution mechanism chooses the desired CFF as between 50-$55Hz$. Then when the recorded video converts the $60Hz$ flicker down to a lower frequency, *e.g.*, $40Hz$, it will be below the CFF. To obtain a larger space for amplitude modulation, we design the block size as about 10° with respect to that human's vertical field of view at about 120°. Hence, based on Figure 32, the normalized amplitude

of the flicker could be as large as 0.2. For each original frame, we add the block-pattern flicker pair to its second and third sub-frames. Figure 36 shows an example of illuminance pollution.

5.3.2 Chromatic Frame Decomposition

After the illuminance frame pollution, the illuminance of each pixel in the sub-frames is determined. For a fusion pair, any pair of corresponding pixels have an infinite number of chromaticity combinations to achieve the desired mix color. We propose to choose a set of combinations that will (approximately) maximize the potential color distortion and spatial deformation.

Given a pixel pair P_1 with $C_1 = (x_1, y_1, Y_1)$ and P_2 with $C_2 = (x_2, y_2, Y_2)$ from a fusion pair, to maximize the recorded color distortion, we need to find out the relation between the distortion and the choice of (x_1, y_1) and (x_2, y_2). The correctly fused color is C with coordinate $(x, y) = \alpha(x_1, y_1) + (1 - \alpha)(x_2, y_2)$, here $\frac{Y_1}{Y_2} = \frac{\alpha}{1-\alpha}$. Recall that color distortion happens when the camera fails to capture the complete light signal, which causes the recorded illuminance ratio of P_1 and P_2 deviates from the correct ratio. Let the recorded ratio be $\frac{Y_1'}{Y_2'} = \frac{\beta}{1-\beta}$, then we have $\alpha \neq \beta$. The distorted color is C' with $(x', y') = \beta(x_1, y_1) + (1 - \beta)(x_2, y_2)$. Then the color distortion is

$$D_c(C, C') = |\alpha - \beta| D_c(C_1, C_2).$$

Here α is determined by the original video and the illuminance pollution, while β is determined by the camera's parameters. This shows that, larger $D_c(C_1, C_2)$ can lead to severe potential color distortion. As a result, when we choose two decomposed colors, we need to maximize the distance between them. Note that, the distance is bound by the range

of the RGB triangle in the CIE diagram (as shown in Figure 30). Three vertexes of the RGB triangle are $R = (0.64, 0.33)$, $G = (0.177, 0.712)$ and $B = (0.15, 0.06)$.

Given the color of the original pixel with color $C = (x, y, Y)$, Y_1 and Y_2 of the decomposed pixels's colors C_1 and C_2 are predetermined. Then determination of (x_1, y_1) and (x_2, y_2) is an optimization problem:

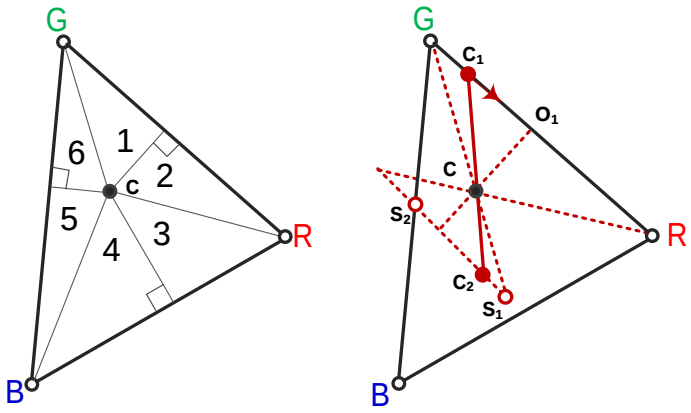
$$\max D_c(C_1, C_2) \quad \text{such that} \tag{8}$$

$$\begin{cases} \frac{D_c(C_1,C)}{D_c(C_2,C)} = \frac{Y_2}{Y_2} \\ \\ \text{both } C_1 \text{ and } C_2 \text{ are within the RGB triangle.} \end{cases}$$

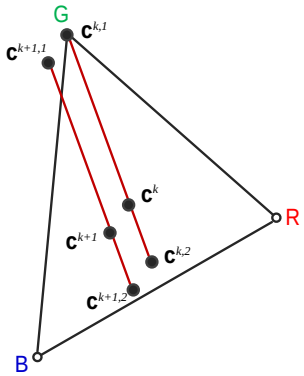
We notice that, the optimal solution must have at least one decomposed color lying on the edge of the RGB triangle. We then propose an algorithm to achieve the optimum with constant time complexity. Our algorithm works as follows. We first divide the RGB triangle into six regions as illustrated in the left figure in Figure.37(a). Then we start to search the local optimum within each region. Within a single region, we find that the optimization objective $D_c(C_1, C)$ changes monotonically (as illustrated in the right figure in Figure.37(a)). Leveraging the monotonicity, one can simply find the optimum in each region using constant computation. We get at most six local optimal solutions. In some regions, there could be no solution. Finally, comparing those six solutions gives us the optimal solution. There is a special case that, three primary colors (red, green and blue) cannot be decomposed. So the primary color remains the same in sub-frames.

5.3.3 Maximize Spatial Deformation

When determining the decomposed color pair, it is difficult to achieve the tradeoff between maximization of color distortion and spatial deformation. Notice that, unlike visual

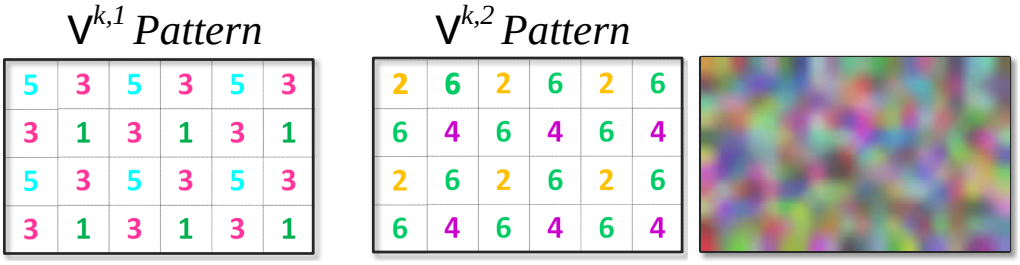


(a) Color space division and solution search



(b) Pixel decomposition cost

Figure 37: Color space decomposition



(a) Patterned Color Selection (b) Mixture

Figure 38: Spatial deformation.

cryptography, we cannot pick an arbitrary sub-frame and then compute the other sub-frame accordingly s.t. the perceived visual effect is exactly same as the original frame, because the constraints on color additive rule. We propose several simple light-weight methods for

achieving a good balance between temporal and spatial quality degradation.

(1) Patterned Color Selection:

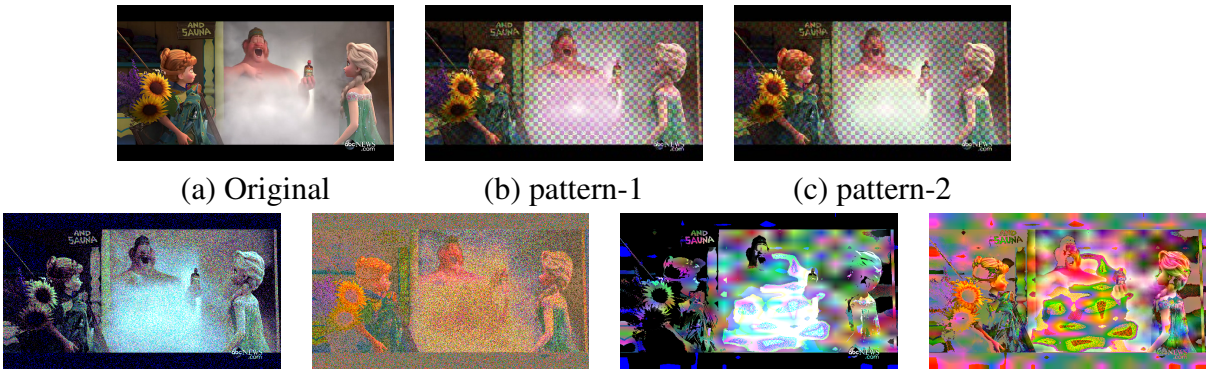
This is based on the 6 regions divided in Figure 37, We divide each sub-frame into small grids of certain fixed size. The grid division can be consistent with the illuminance pollution. For one sub-frame of a fusion pair, we assign each grid a region following the left pattern presented in Figure 38. If a pixel with color C is in a grid labeled Region q, then the decomposed color C_1 will only be searched within the Region q. Thus, the whole sub-frame will have the assigned pattern despite the original shape. The other sub-frame of this fusion pair will use the right pattern in Figure 38(a) for finding the corresponding color to produce the original color. Figure 39(b) and (c) illustrate two sub-frames produced for an original frame in Figure 39(a).

(2) Random Color Selection:

For each pixel, we first select a random color, and then compute the corresponding pixel's color in the complement frame. As shown in Figure 39, although intuitively it will produce a pair of random sub-frames, the actual produced sub-frames still contains a rich shape information. The reason is that the similar color in adjacent pixels may result in similar optimal solution in each region. Figure 39(d) and (e) show two sub-frames produced.

(3) Mixture of Random and Smoothing:

The third method is to use a combination of random choice for each pixel and smoothing among adjacent pixels. First, for each pixel with color C, we randomly select a color C_1 till that there is another color C_2 such that C is produced using color additive rule with C_1 and C_2. Then for each pixel $P_{i,j}^k$, the color of the pixel $P_{i,j}^{k,1}$ in the first sub-frame is an average of the neighboring pixels in this sub-frame. Figure 38(b) shows an example of chromatic



(a) Original (b) pattern-1 (c) pattern-2

(d) random-1 (e) random-2 (f) mixed-1 (g) mixed-2

Figure 39: Encoding the subframes for deforming and hiding the spatial information in the frame.

deformation map for the mixture based method. Figure 39(f) and (g) show two sub-frames produced.

Figure 39 illustrates the original frame, and a pair of sub-frames produced by these different methods. Note that such randomization and mixture can effectively remove the spatial information in the original frame with different tradeoffs between the viewing experience and anti-piracy ability.

5.3.4 Reducing the Encoding Cost

The computational overhead for decomposing the frames of original video into two successive sub-frames pixel by pixel cannot be neglected, especially for high-definition video with 1920×1080 spatial resolution. Leveraging the inherent property of the video, we improve the method to reduce the decoding overhead. Given a normal $30fps$ video, the color for the pixel P_{ij} in both k and $k+1$ frame are $C_{ij}^k = (x_{ij}^k, y_{ij}^k, Y_{ij}^k)$ and $C_{ij}^{k+1} = (x_{ij}^{k+1}, y_{ij}^{k+1}, Y_{ij}^{k+1})$ respectively. Thus the color difference,

$$D_c(C_{ij}^k, C_{ij}^{k+1}) = \sqrt{(x_{ij}^k - x_{ij}^{k+1})^2 + (y_{ij}^k - y_{ij}^{k+1})^2},$$

between the same pixel in two successive original video frames could be considered as the chromatic distance in the color space. To reduce the computational overhead, we define ϵ as the threshold of the color difference, and if the difference is less than ϵ the pixel color in the two sub-frame could be calculated directly from the previous one without conducting the color distortion maximization repeatedly.

We use a two stage method of complementary color pair determination as illustrated in Figure 37(b): (1) we draw a parallel line segment to the line segment $C_{ij}^{k,1}C_{ij}^{k,2}$ in the previous frame. (2) the illuminance of pixels in the sub-frames are determined after the illuminance frame pollution. Then we shrink the length of the parallel line segment align with the illuminance ratio so that the adjusted line segment is within the RGB triangle, and determine the coordinate of both $(x_{ij}^{k+1,1}, y_{ij}^{k+1,1})$ and $(x_{ij}^{k+1,2}, y_{ij}^{k+1,2})$ with maximized line segment distance between $C_{ij}^{k+1,1}$ and $C_{ij}^{k+1,2}$.

5.4 Evaluation

In this section, I will present the performance evaluation of KALEIDO via experiments. The prototype of KALEIDO is implemented in C++ with OpenCV library. KALEIDO re-encodes the original video stream into high frame rate video stream, and displays it through regular LCD monitors or projectors. We then evaluate the video quality of both watch-only video and pirate video respectively. All the watch-only video are generated through the basic methods mentioned in the previous section, and compare the corresponding pirate video captured by multiple cameras with original video clips to determine whether the content of the original video is protected through some standard video quality assessment metrics. As such objective video quality metrics may not directly reflect the subjective viewing experience by human eyes, we also combine both objective and subjective experiments to

measure the effectiveness of the pirate video recording prevention.

5.4.1 Experiment Settings

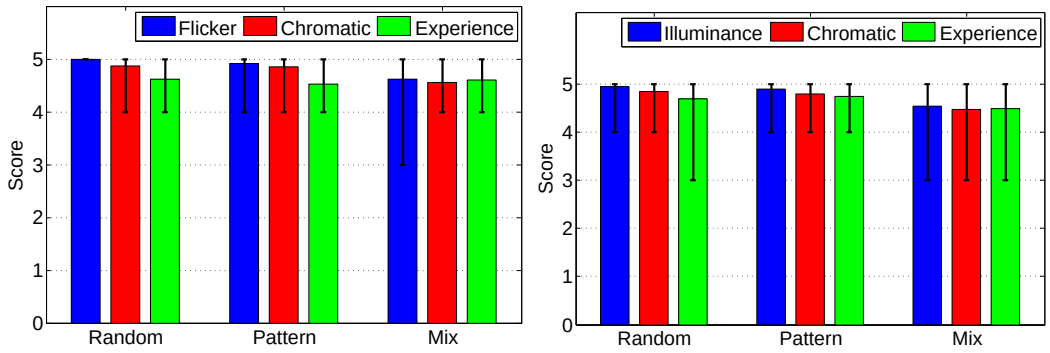
In our evaluation, we use both LCD monitor and projector as the main display. The 27' LCD monitor (AOC G2770PQU) supports 1920×1080 spatial resolution and up to 144Hz refresh rate, while Acer $D600$ projector supports 1280×720 spatial resolution with 120fps frame rate. During the evaluation, we set the frame rate for both two display devices as 120fps. We simulate the working scenarios of both movie and presentation, and verify whether the viewing quality of watch-only video escapes from degradation. On the camcorder side, we use 5 different smartphones (iPhone 5s, iPhone 6, Samsung Note3, Note4 Edge, HTC M8) to capture and record the video projected on the screen.

We also employ a various of video source to examine whether our system could be widely applicable. We select twenty different high-definition (1280×720) video clips with different characteristics on brightness, contrast, and motion. The content of videos are ranged from drama, sports, landscape to animation.

The subjective perception quality of is conducted through users' study. We invite 50 volunteers in aging from 20 to 40 with 31 males and 19 females. All the volunteers have regular visual sensitivity, and one of them is graphic designer with great sensitivity to the video quality.

5.4.2 Watch-Only Video Quality Assessment

We first evaluate the video quality of the displayed watch-only video. As the original video is re-encoded before displaying, and the content in each frame in the watch-only video is irregular and random, it is difficult to evaluate the quality of the watch-only video objectively by existing standard quality assessment metrics. Thus, we only evaluate its



(a) LCD monitor (b) Projector

Figure 40: Average subjective score.

quality by subjective watching experience of volunteers.

In the evaluation, we display both the original video and the watch-only video side by side in two identical display system, and ask them to rate the video quality of the watch-only video by comparing with the original one in three aspects respectively: illuminance level, chromatic correctness, and overall watching experience. Similar to [106], we use score 5 to 1 for each aspect, where 5 indicates the highest quality without any differences in illuminance, chromaticity and the video quality is satisfying; 4 represents the difference being "almost unnoticeable", and 3 to 1 denote "merely noticeable", "evident noticeable", and "strong noticeable or artifact". Since the format of all the selected video clips are high-definition, only scores above 4 indicate the acceptable video quality. We collect the watching experience feedback from all volunteers, and plot the average score in Figure 40. All the watch-only videos show great smoothness in both projector and LCD monitor, where no jitter is noticed by the audiences. The main subjective difference come from flickers brought by both the illuminance change and the chromatic distortion, which also results in the spacial deformation. The encoding method with random choice of pixel col-

ors provides the best view quality, where the average scores for the first two metrics are both greater than 4.9. 96% volunteers did not even notice they are watch-only video clips. The encoding method with pattern follows with slight drop in performance, because the illuminance and chromatic flicker blocks have larger size than pixels. 92% volunteers did not distinguish them. The encoding method with mixed techniques disturb the original frames mostly, where audiences may experience distortion of both chromaticity and illuminance. Although 38% volunteers noticed that those video clips are re-encoded, but the degradation is acceptable and the average score is above 4.

We also consider other parameters affecting the watching experience, including display devices, different light conditions and different video types. As shown in Figure 40, LCD monitors have a slightly better performance than projectors, possibly because the projector has a larger display area, which makes illuminance and chromatic flicker easily noticable. Moreover, light condition and video type do not cause significant differences of watching experience.

5.4.3 Pirated Video Quality Assessment

I will then evaluate the performance of our KALEIDO prototype in dealing with piracy camcorder by comparing the pirate video first with the original video clips to present the quality degradation of the pirate watch-only video. However, it is still not easy to determine whether the large amount of quality degradation results from the recording process or the success of frame decomposition. Since multiple factors will lead to quality degradation, and the standard metrics for video quality assessment do not have strong linear correlation to the actual watching experience, it is difficult to compare the definite video quality based on the metric results only. Essentially, the content of the pirate video from regular video is

easy to recognize, especially when the recording devices are increasingly powerful. Here we also compare the quality of pirated watch-only video to the pirated video from the original video.

Five smartphones are used to record pirate video, where the capturing rate is $1080p$ in $30fps$ or $60fps$ and $720p$ in $120fps$. The extensive evaluation is conducted in an indoor office with two different light conditions: *nature light condition* representing the presentation scenarios and *dark condition* indicating the theater scenarios. The quality of the pirate video will be evaluated both subjectively by watching experience and objectively by standard metrics.

5.4.3.1 Subjective Viewing Experience

Several facts could affect the pirate video quality, including display device (LCD or projection), camera capture frame rate and light condition. Figure 41 illustrates the comparison of pirate videos captured using different display devices or fps, where the top-left image is the snapshot of the original video frame. If the pirate video is recorded from the playing of original video, it could still reveal most significant detail of the image, as shown in top-right. When recorded from watch-only video, the content of the frames is difficult to recognize compared with from the original video. For example, the middle two frames come from the watch-only video played in LCD and the last two are displayed by projector. We notice that the pirate video quality degrades with increasing fps, because the flicker frequency down-conversionand more unstable frame interval. Different display devices and light conditions do not cause any significant difference of watching experience. One thing we should keep in mind is that although some of the frames still could be perceived by human eyes, when a sequential of such distorted image frames are played in a

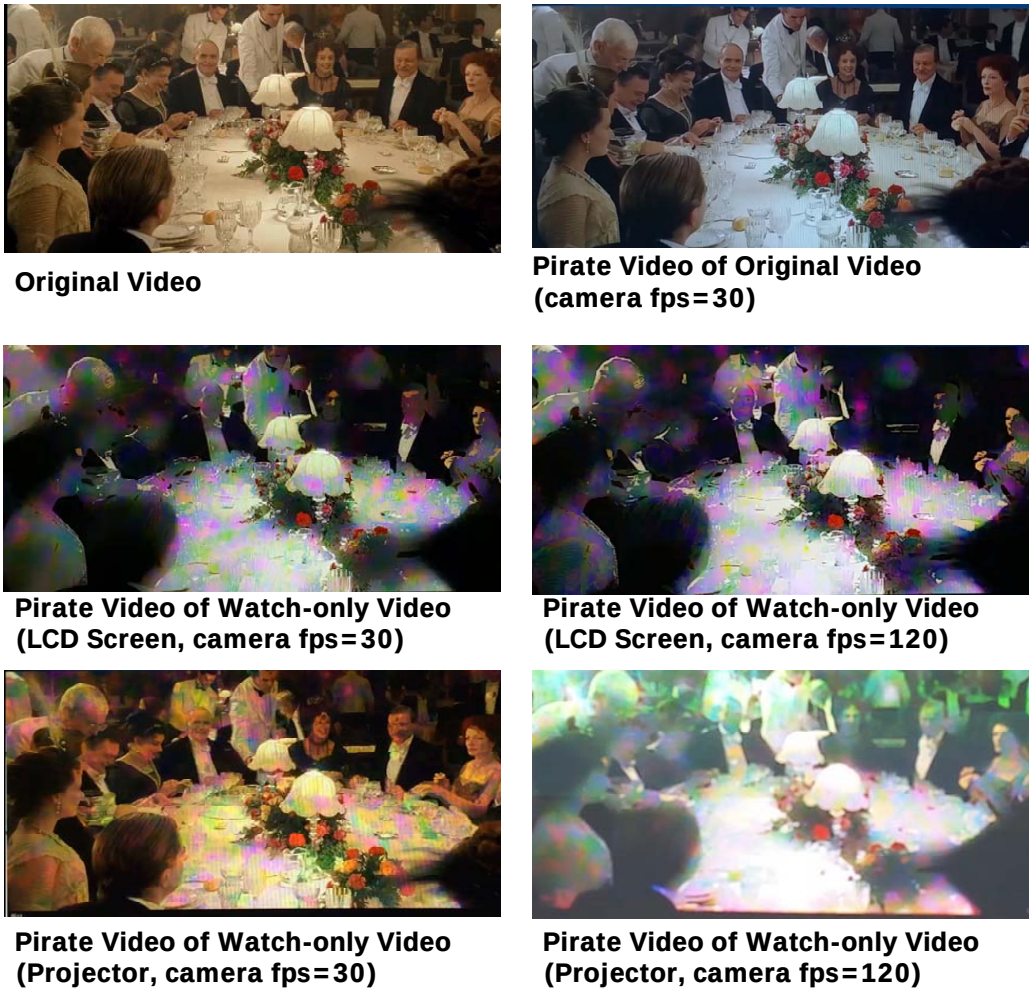
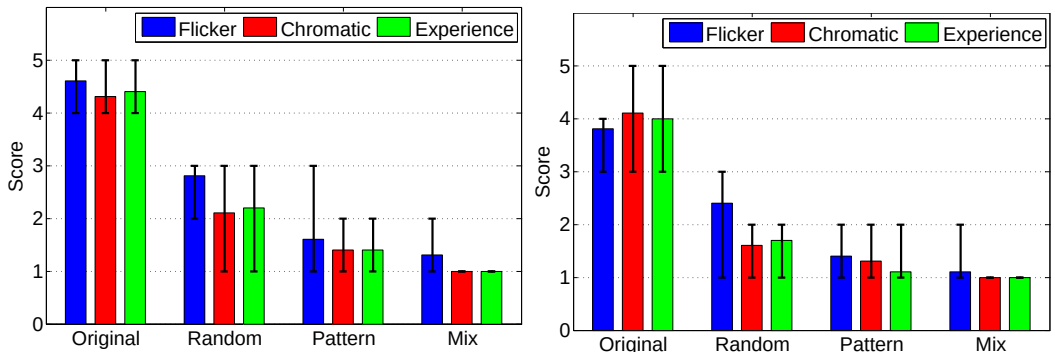


Figure 41: The snapshot of pirate videos with different capturing scenarios.

regular frame rate, the viewers' viewing experience is significantly affected when playing the pirate video recorded from the watch-only video. Thus, we pay more attention to the overall video quality degradation.

In the subjective assessment, we consider the content of the pirated video, compared to both original video clips and pirated original video clips. We display the original video, pirated video of original video and pirate video of watch-only video side by side in three identical display systems. The rating score for the pirated video is still from 1 to 5 as in previous evaluation. Figure 42 illustrates the rating for all pirated video clips captured using



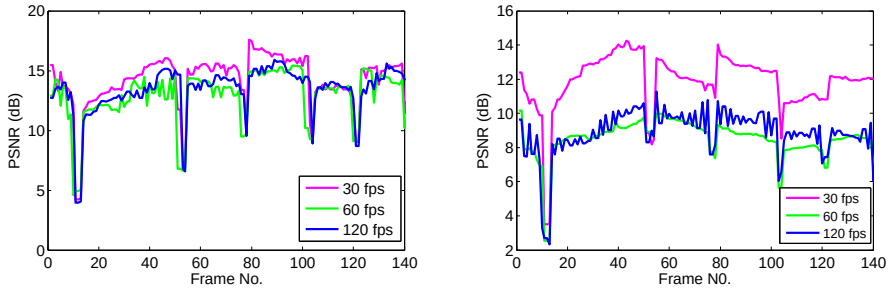
(a) LCD monitor (b) Projector

Figure 42: Subjective view experiences: pirate original video, pirate watch-only video with various techniques.

different display devices (LCD monitor and projector) respectively. The score for pirated original video is about 4, which indicates acceptable quality. Both our watch-only video effectively reduces the quality of the pirate video from the watch-only video in all tested scenarios. 96% volunteers claim the quality degradation is intolerable, and the average rating score is below 2.

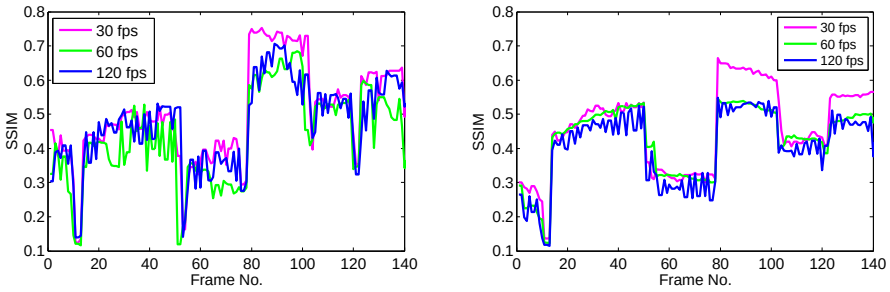
5.4.3.2 Objective Measurement

We use five different standard metrics to measure the quality of the pirated video, including PSNR, SSIM, CD, and Histogram. For objective measurement, we setup the evaluation scenario to the finest where the video is being displayed in the screen with largest brightness and the camera is directly facing the screen so that the whole screen could be captured without trapezoid. The usual pirate videotaping scenario would be worse than this ideal testing scenario. Thus, if the quality of the pirate video in this ideal scenario is intolerable, the pirate video taken in worse conditions will experience more severe quality degradation. Due to the disparity between the frame rate of the original video ($30fps$) and the pirate video, we duplicate the frame of original video to align each frame to the captured frames



(a) LCD monitor (b) Projector

Figure 43: PSNR in different recording frame rates.



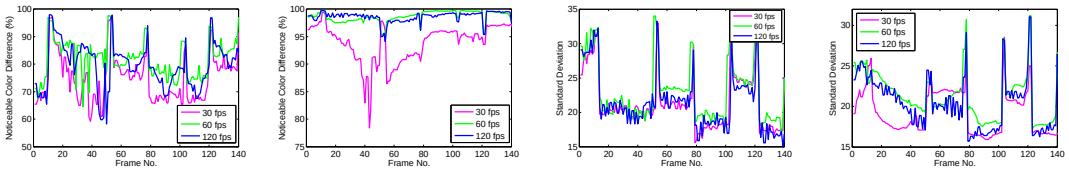
(a) LCD monitor (b) Projector

Figure 44: SSIM in different recording frame rates.

in the pirate video.

PSNR (Peak-Signal-to-Noise-Ratio) is one of the most basic video quality assessment metrics to measure the quality of lossy video compression so as to provide an approximation to human perception of re-encoded video quality. Figure 43 plots the real-time PSNR for a random selected video clip in different shooting frame rates. The PSNR usually has a value ranging form 30 to 50dB for medium to high quality video [107]. However, the PSNR values fluctuate in a wide range for all pirate video frames, and the values are always below 18dB, indicating a significant quality degradation.

SSIM [108] is proposed as a method to calculate the similarity between two images. The SSIM gets the best value of 1 for two identical images, and with the quality decrease, the

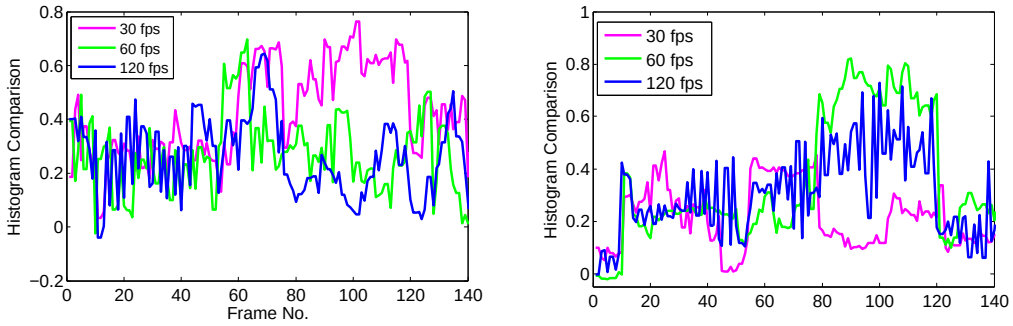


(a) LCD monitor (b) Projector (c) LCD monitor (d) Projector

Figure 45: Color difference in different recording frame rates (proportion and standard deviation).

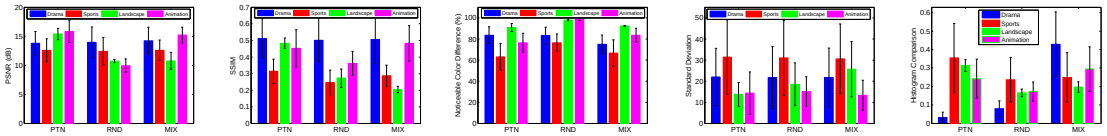
value of SSIM drops accordingly. The value drops below 0.7 when the image contain large distortion, and the content is difficult to recognized clearly [108]. Our evaluation (Figure 44) shows that the videos have strong structure distortion when captured by camcorder in three different frame rates from 30fps to 120fps. The average SSIM for LCD are 0.4615, 0.4305 and 0.4070 respectively and 0.5053, 0.4339 and 0.4717 for the projector.

Color Difference (CD, or Chromatic Aberrations CA) is another reliable metric to verify the quality of captured video stream, which usually is generated from a failure of lens to focus all colors to the convergence point. Recording a pirate video will definitely generate color difference, and the value of the color difference determines the amplitude of the color distortion. In this case, we adopt ICEDE2000 [72] to calculate the color difference between the pirate video frame and original video frame on each pixel. When the value of ICEDE2000 exceeds 6, the color difference could be noticed clearly. As the amplitude of color difference usually has nonuniform distribution on the frame, it is ineffective to measure the average color distortion. Instead we calculate the proportion of pixels in each frame that has color difference larger than 6, and compute the variance for those pixels. Based on our evaluation (Figure. 45), such proportion is beyond 70% for most of the video frames and the standard deviation for the color difference is over 21.



(a) LCD monitor (b) Projector

Figure 46: Histogram in different recording frame rates.



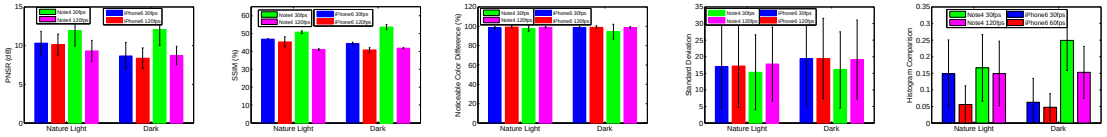
(a) PSNR (b) SSIM (c) CD proportion (d) CD std. Dev. (e) Histogram

Figure 47: The quality evaluation for different decomposition methods.

We also compare the color histogram of the pirate video to the original video, and plot the correlation for different frame rates in two display systems. As shown in Figure 46, the value of histogram has no obvious correlation to the frame rate, and all the video shows moderate correlation.

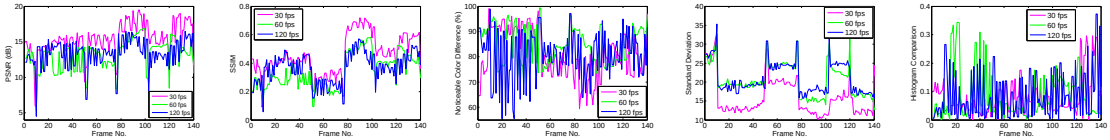
We then evaluate the performance of quality degradation in different decomposition methods (pattern based, random, and mixed) (see Figure 47). Clearly, all our methods distort the color of original frames, which leads to significant quality degradation in all videos.

We extend our comparison of the video quality degradation in two difference light conditions. The watch-only video is displayed by projector, which is used to simulate the theater and presentation scenarios, and we record the video by two most popular mobile phones



(a) PSNR (b) SSIM (c) CD proportion (d) CD std. Dev. (e) Histogram

Figure 48: The video quality assessment in two light conditions.



(a) PSNR (b) SSIM (c) CD proportion (d) CD std. Dev. (e) Histogram

Figure 49: Comparing pirate video for both original and watch-only.

in two frame rates. As the results shown in Figure 48, the pirate video contains similar quality degradation in both environments, and when the camcorder captures the video in lower frame rate, the amplitude of quality degradation is lower than high frame rate.

The purpose of the previous experiments is to present the quality degradation of the pirate watch-only video, comparing the original video clips. We now evaluate the same metrics of pirated watch-only video compared with pirated original video (video recorded from playing the original video). As shown in Figure 49, our results indicate that the pirate watch-only video also has severe quality degradation compared to the pirate original video of non-modified version. Since the pirate original video has already given viewers a unpleasant watching experience, the pirate watch-only video has a much worse quality.

5.5 Summary:

KALEIDO re-encodes the original video into a watch-only one. The subjective assessment shows that the watch-only video can preserve the viewer's watching experience satisfactorily. And both the subjective and objective evaluation results indicate that the quality

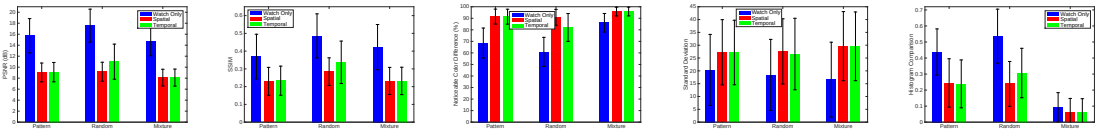
of the pirate video from watch-only video is severely degraded in all cases (different combinations of display device, camera fps, video type and light condition) compared to both original video and pirated original video. There is still a room for audience experience optimization, and we will improve it in our future work.

5.6 Discussion and Open Issues

KALEIDO is a first step towards solving piracy problem by generating watch-only videos. While the evaluations demonstrate that KALEIDO is promising, there are some limitations and open problems as discussed below.

5.6.1 System Applicability:

In our system design, we leverage the rolling shutter effect to achieve watch-only video against mobile devices. Thus our method may not be working well when facing high-end cameras with global shutter. But, our mechanism could prevent most piracy events caused by current consumer cameras, which is the main focus of this work. On one hand, pirate video captured by personal mobile devices cause great loss to movie industry and severe infringement of copyright. It is easy to prevent high-end professional camcorder from cinema or lecture hall, but it's difficult to forbid attendees to bring personal mobile phones. In MPAA's latest attempt to crack down on piracy, it is pressuring movie theaters to adopt a ban on mobile phones with cameras and certain kinds of eyeglasses, which causes great concern on the security of personal phones and degrades the experience of audiences. On the other hand, the rolling shutter camera dominates the consumer camera market. According to Grand View Research's report about image sensor market [**?**], by 2013 CMOS image sensors takes 83.8% market share, while CCD image sensors takes only 16.2%. By 2015, CMOS shipments will amount to 3.6 billion units or 97 percent market



(a) PSNR (b) SSIM (c) CD proportion (d) CD std. Dev. (e) Histogram

Figure 50: The video quality comparison after noise removal

share, compared to CCD shipments of just 95.2 million, or 3 percent [**?**]. The majority of CMOS sensors found in the consumer market utilize a rolling shutter, while only few expensive high-end CMOS sensor can support global shutter, as the global shutter is hard to accomplish in current CMOS designs. We will explore the solution against high-end cameras in our future work.

5.6.2 Post Processing:

Since KALEIDO re-encodes the common video into a watch-only one, one of possible attack to our approach is to remove the noise in the video through post processing. Generally, video denoising methods have two different categories: spatial and temporal. Non-local means are the most common spatial video denoising method, which removes the noise at a pixel through certain operations with neighbors within single video frame, such as gaussian weighted average. Although temporal approaches will reduce the noise between frames through tracking blocks along trajectories defined by motion vector and removing the noise of a pixel by taking a number of same pixels from different frames, it is still not suit our watch-only video. In our method we decompose each frame by chromaticity and illuminance in a random manner, and pollute frame temporally and deform frames spatially. Due to the rolling shutter effect and unstable inter-frame intervals, there are information loss and distortion in the recorded frames rather than simple Gaussian white

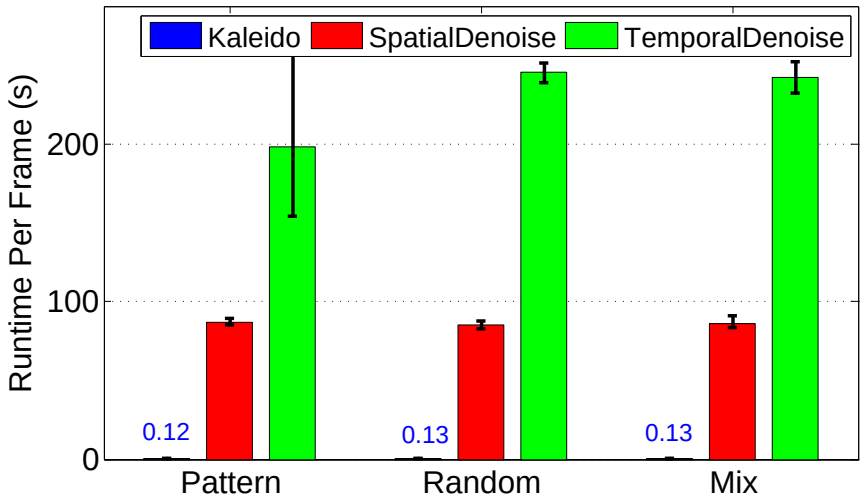


Figure 51: Processing runtime of one video frame for generating watch-only video and denoising. It takes 0.12s for KALEIDO to process one frame.

noise, and our techniques maximize such loss and distortion. Therefore, it is still difficult to restore original pixels using incomplete and distorted information. To better present that our method is resistent to existing noise removal techniques, we conduct the attack through two mainstream video denoising method: spatial and temporal noise cancelation process, and plot the results in Figure 50. In this experiment, we compared all the processed videos to the pirate video of original video with standard metrics as before, and we also put the video quality metrics of the watch-only video as comparison. Obviously, the common post denoising process not only cannot recover the original video, but also deteriorate the video quality compared to the watch-only version in all five basic metrics due to recognizing noise incorrectly. Therefore, KALEIDO is insusceptible to common denoising attack, and guarantees the reliable video privacy preservation.

5.6.3 System Overhead:

KALEIDO does not generate watch-only video while playing it, but converts the video off-line and loads processed frames to GPU buffer for playing to optimize the watch experi-

ence. We evaluate the computation cost for video conversion using a commercial computer with Intel i7-4790 3.6GHZ CPU and 8G RAM. For software, we employ matrix operations provided OpenCV to achieve optimal performance. As presented in Figure 51, it takes 0.12s in average to process one 1280×720 video frame, *i.e.*, the process speed is about $8.3fps$. Compared to generating watch-only video, denoising is orders of magnitude slower. Two mainstream video denoising methods cost about 85s and 200s to process one frame separately. For the storage cost, since our method increase the frame rate from $30fps$ to $120fps$, the watch-only video quadruples the file size of the original one.

5.6.4 Watching Experience Degradation:

Although KALEIDO can severely reduce the quality of pirate video, we admit that there still has some degradation on watching experience for onsite audiences. Actually, there is a tradeoff between watching experience and piracy prevention. We design our method to maximize the viewing experience and the evaluation results show that the degradation is nearly negligible. There is still a room to improve the viewers' experience, and we will leave this as our future work.

5.7 Summary

In this chapter I propose a scheme for re-encoding the original video frames such that it can prevent a good quality pirate videotaping of the displayed video in the public using commercial off-the-shelf smart-devices, while do not affect the high-quality viewing experience of live audiences. I design exploits the subtle disparities between the screen-eye link and the screen-camera link. Extensive evaluations of our implementation demonstrate its effectiveness against pirate video. One remaining work is to improve the encoding efficiency, and reduce the time delay of generating the watch-only video. A more daunting

challenge is to design a scheme that can even prevent a good-quality pirate video-taping by high-end professional cameras.

CHAPTER 6: CONCLUSION

In this dissertation, we have studied the problem of privacy preserving mechanism in mobile computing and networking. We have designed new solutions to distinguish the identity of user based on the behavioral biometrics so as to protect the private date in the local device. We also discussed the different possible solutions to deal with the privacy leakage when sharing and broadcasting photos and videos in the public. We briefly summarize our completed work and the future work.

We have made the following contributions:

- We provided an impersonation-proof authentication method, which exploits the fact that biometric features are difficult to replicate or imitate.
- We proposed multi-level features which is used to build a consistent and robust human behavioral model with extremely high distinguishability.
- Besides the obliviousness, the proposed framework (*SilentSense*) has continuity and adaptiveness during the authentication. It continuously detects whether the device operator is the owner when someone is using the device, and it also adaptively update the authentication model when the user's behavior pattern changes.
- We proposed a new privacy protection paradigm that aims to give the privacy control back to the people being photographed.

- We analyzed the whole life-cycle of a photo and identify that the Photo Service Providers are the best place to exert privacy protection. And we design and implement a Privacy Expressing and Respecting Protocol (PERP) and a Privacy.Tag concept.

- By taking the advantage of the limited disparities between the screen-eye channel and the screen-camera channel, we proposed and developed a light-weight system to prevent unauthorized users from video taping a video played on a screen, such as theater, while the viewing experience for live audience are note affected.

- The proposed system does not require extra hardware and is purely based on re-encoding the original video frame into multiple frames used for displaying.

REFERENCES

[1] Apple iphone fingerprint reader confirmed as easy to hack. goo.gl/O15eO7.

[2] Camera phone predator alert act. govtrack.us/congress/bills/111/hr414#citations/.

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

[4] Google glass. google.com/glass/start/.

[5] Google street view. google.com/streetview.

[6] Google street view privacy and security. google.com/maps/about/behind-the-scenes/streetview/privacy/.

[7] How many photos are uploaded to flickr every day, month, year? goo.gl/hDYZSQ.

[8] Instagram press center. instagram.com/press/.

[9] Japanese iphone makes loud shutter sound in silent mode. goo.g/dPo7SZ.

[10] Memoto. memoto.com/.

[11] Microsoft research face sdk. research.microsoft.com/en-us/projects/facesdk/.

[12] Picasa. picasa.google.com/.

[13] Preventing movie piracy. goo.gl/OJPozs.

[14] Privacy international. goo.g/kFKlqe.

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

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

[17] Visualead. visualead.com/.

[18] The web robots pages. robotstxt.org/.

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

[20] S. Abdulla. New visual cryptography algorithm for colored image. *arXiv preprint arXiv:1004.4445*, 2010.

[21] E. Achtert, H.-P. Kriegel, and A. Zimek. Elki: A software system for evaluation of subspace clustering algorithms. In *SSDBM*, pages 580–585, 2008.

[22] A. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *Privacy enhancing technologies*, pages 36–58. Springer, 2006.

[23] 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*, volume 7, pages 357–366, 2007.

[24] S. J. Anderson and D. C. Burr. Spatial and temporal selectivity of the human motion detection system. *Vision research*, 25(8):1147–1154, 1985.

[25] P. Andriotis, T. Tryfonas, G. Oikonomou, and C. Yildiz. A pilot study on the security of pattern screen-lock methods and soft side channel attacks. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 1–6. ACM, 2013.

[26] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49–60, 1999.

[27] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. *WOOT*, 10:1–7, 2010.

[28] 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.

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

[30] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 321–334. IEEE, 2007.

[31] H. Blasinski, O. Bulan, and G. Sharma. Per-colorant-channel color barcodes for mobile applications: An interference cancellation framework. *Image Processing, IEEE Transactions on*, 22(4):1498–1511, 2013.

[32] C. Bo, G. Shen, J. Liu, X.-Y. Li, Y. Zhang, and F. Zhao. Privacy. tag: Privacy concern expressed and respected. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 163–176. ACM, 2014.

[33] C. Bo, L. Zhang, T. Jung, J. Han, X.-Y. Li, and Y. Wang. Continuous user identification via touch and movement behavioral biometrics. In *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International*, pages 1–8. IEEE, 2014.

[34] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[35] K. Chinomi, N. Nitta, Y. Ito, and N. Babaguchi. Prisurv: Privacy protected video surveillance system using adaptive visual abstraction. In *Advances in Multimedia Modeling*, pages 144–154. Springer, 2008.

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

[37] N. Clarke, S. Karatzouni, and S. Furnell. Flexible and transparent user authentication for mobile devices. In *Emerging Challenges for Security, Privacy and Trust*, pages 1–12. Springer, 2009.

[38] N. L. Clarke and S. Furnell. Authenticating mobile phone users using keystroke analysis. *International Journal of Information Security*, 6(1):1–14, 2007.

[39] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 243–246. IEEE, 1996.

[40] H. Crawford, K. Renaud, and T. Storer. A framework for continuous, transparent mobile device authentication. *Computers & Security*, 39:127–136, 2013.

[41] F. Dufaux and T. Ebrahimi. Scrambling for privacy protection in video surveillance systems. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1168–1174, 2008.

[42] F. Dufaux and T. Ebrahimi. A framework for the validation of privacy protection solutions in video surveillance. In *ICME*, pages 66–71, 2010.

[43] J. Fan, H. Luo, M.-S. Hacid, and E. Bertino. A novel approach for privacy-preserving video sharing. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 609–616. ACM, 2005.

[44] F. Farrell. Fitting physical screen parameters to the human eye. *Vision and visual dysfunction: The man-machine interface. Boca Raton, FL: CRC*, 1991.

[45] T. Feng, X. Zhao, B. Carbunar, and W. Shi. Continuous mobile authentication using virtual key typing biometrics. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 1547–1552. IEEE, 2013.

[46] J. Frank, S. Mannor, and D. Precup. Activity and gait recognition with time-delay embeddings. In *AAAI*. Citeseer, 2010.

[47] D. Gafurov, K. Helkala, and T. Søndrol. Biometric gait authentication using accelerometer sensor. *Journal of computers*, 1(7):51–59, 2006.

[48] G. M. Gilbert. Dynamic psychophysics and the phi phenomenon. *Archives of Psychology (Columbia University)*, 1939.

[49] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. ACM, 2006.

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

[51] J. Häkkilä and J. Mäntyjärvi. Developing design guidelines for context-aware mobile applications. In *Proceedings of the 3rd international conference on Mobile technology, applications & systems*, page 24. ACM, 2006.

[52] T. Hao, R. Zhou, and G. Xing. Cobra: color barcode streaming for smartphone systems. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 85–98. ACM, 2012.

[53] Y.-C. Hou. Visual cryptography for color images. *Pattern Recognition*, 36(7):1619–1629, 2003.

[54] S. Hranilovic and F. R. Kschischang. A pixelated mimo wireless optical communication system. *Selected Topics in Quantum Electronics, IEEE Journal of*, 12(4):859–874, 2006.

[55] W. Hu, H. Gu, and Q. Pu. Lightsync: Unsynchronized visual communication over screen-camera links. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 15–26. ACM, 2013.

[56] W. Hu, J. Mao, Z. Huang, Y. Xue, J. She, K. Bian, and G. Shen. Strata: layered coding for scalable visual communication. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 79–90. ACM, 2014.

[57] W. Huang and W. H. Mow. Picode: 2d barcode with embedded picture and vicode: 3d barcode with embedded video. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 139–142. ACM, 2013.

[58] S.-s. Hwang, S. Cho, and S. Park. Keystroke dynamics-based authentication for mobile devices. *Computers & Security*, 28(1):85–93, 2009.

[59] Y. Jiang, K. Zhou, and S. He. Human visual cortex responds to invisible chromatic flicker. *Nature neuroscience*, 10(5):657–662, 2007.

[60] R. Joyce and G. Gupta. Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176, 1990.

[61] T. Jung, X.-Y. Li, Z. Wan, and M. Wan. Privacy preserving cloud data access with multi-authorities. In *INFOCOM, 2013 Proceedings IEEE*, pages 2625–2633. IEEE, 2013.

[62] T. Jung, X. Mao, X.-Y. Li, S.-J. Tang, W. Gong, and L. Zhang. Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation. In *INFOCOM, 2013 Proceedings IEEE*, pages 2634–2642. IEEE, 2013.

[63] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, and A. LaMarca. Ensemble: cooperative proximity-based authentication. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 331–344. ACM, 2010.

[64] A. K. Karlson, A. Brush, and S. Schechter. Can i borrow your phone?: understanding concerns when sharing mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1647–1650. ACM, 2009.

[65] H. Khan, A. Atwater, and U. Hengartner. Itus: an implicit authentication framework for android. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 507–518. ACM, 2014.

[66] K. Killourhy and R. Maxion. Why did my detector do that?! In *Recent Advances in Intrusion Detection*, pages 256–276. Springer, 2010.

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

[68] J. Koreman, A. Morris, D. Wu, S. Jassim, H. Sellahewa, J. Ehlers, G. Chollet, G. Aversano, H. Bredin, S. Garcia-Salicetti, et al. Multi-modal biometric authentication on the securephone pda. 2006.

[69] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Cell phone-based biometric identification. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–7. IEEE, 2010.

[70] T. Li, C. An, A. T. Campbell, and X. Zhou. Hilight: Hiding bits in pixel translucency changes. *ACM SIGMOBILE Mobile Computing and Communications Review*, 18(3):62–70, 2015.

[71] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing facebook privacy settings: User expectations vs. reality. In *IMC*, pages 61–70. ACM, 2011.

[72] M. R. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26(5):340–350, 2001.

[73] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero. A fast eavesdropping attack against touchscreens. In *Information Assurance and Security (IAS), 2011 7th International Conference on*, pages 320–325. IEEE, 2011.

[74] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri. Keystroke dynamics authentication for mobile phones. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 21–26. ACM, 2011.

[75] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 2, pages ii–973. IEEE, 2005.

[76] R. J. Mason, R. S. Snelgar, D. H. Foster, J. R. Heron, and R. E. Jones. Abnormalities of chromatic and luminance critical flicker frequency in multiple sclerosis. *Investigative ophthalmology & visual science*, 23(2):246–252, 1982.

[77] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[78] M. Naor and A. Shamir. Visual cryptography. In *Advances in CryptologyłEUROCRYPT'94*, pages 1–12. Springer, 1995.

[79] M. Naor and A. Shamir. Visual cryptography ii: Improving the contrast via the cover base. In *Security protocols*, pages 197–202. Springer, 1997.

[80] E. M. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *Knowledge and Data Engineering, IEEE Transactions on*, 17(2):232–243, 2005.

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

[82] S. D. Perli, N. Ahmed, and D. Katabi. Pixnet: Lcd-camera pairs as communication links. *ACM SIGCOMM Computer Communication Review*, 41(4):451–452, 2011.

[83] A. Poller, M. Steinebach, and H. Liu. Robust image obfuscation for privacy protection in web 2.0 applications. In *IS&T/SPIE Electronic Imaging*, pages 830304–830304. International Society for Optics and Photonics, 2012.

[84] C. Poynton. Frequently asked questions about color. *Retrieved June*, 19:2004, 1997.

[85] M.-R. Ra, R. Govindan, and A. Ortega. P3: Toward privacy-preserving photo sharing. In *NSDI*, pages 515–528, 2013.

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

[87] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 249–260. IEEE Press, 2014.

[88] 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.

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

[90] V. Rijmen and B. Preneel. Efficient colour visual encryption or shared colors of benetton. *rump session of EUROCRYPT*, 96, 1996.

[91] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. Progressive authentication: Deciding when to authenticate on mobile phones. In *USENIX Security Symposium*, pages 301–316, 2012.

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

[93] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In *Information, Security and Cryptology–ICISC 2009*, pages 229–244. Springer, 2010.

[94] F. Schaub, R. Deyhle, and M. Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, page 13. ACM, 2012.

[95] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu. Enabling video privacy through computer vision. *IEEE Security and Privacy*, 3(3):50–57, 2005.

[96] M. Shahzad, A. X. Liu, and A. Samuel. Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 39–50. ACM, 2013.

[97] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. In *Information security*, pages 99–113. Springer, 2011.

[98] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan. Scalable crowd-sourcing of video from mobile devices. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 139–152. ACM, 2013.

[99] A. Studer and A. Perrig. Mobile user location-specific encryption (mule): using your office as your password. In *Proceedings of the third ACM conference on Wireless network security*, pages 151–162. ACM, 2010.

[100] F. Tari, A. Ozok, and S. H. Holden. A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In *Proceedings of the second symposium on Usable privacy and security*, pages 56–66. ACM, 2006.

[101] 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.

[102] 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.

[103] C. W. Tyler and R. D. Hamer. Analysis of visual modulation sensitivity. iv. validity of the ferry-porter law. *JOSA A*, 7(4):743–758, 1990.

[104] T. Vu, A. Baid, S. Gao, M. Gruteser, R. Howard, J. Lindqvist, P. Spasojevic, and J. Walling. Distinguishing users with capacitive touch communication. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 197–208. ACM, 2012.

[105] A. Wang, Z. Li, C. Peng, G. Shen, G. Fang, and B. Zeng. Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication. In *ACM Mobisys*, 2015.

[106] A. Wang, C. Peng, O. Zhang, G. Shen, and B. Zeng. Inframe: Multiflexing full-frame visible communication channel for humans and devices. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, page 23. ACM, 2014.

[107] Y. Wang. Survey of objective video quality measurements. 2006.

[108] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19(2):121–132, 2004.

[109] R. B. Wolfgang and E. J. Delp. A watermark for digital images. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 219–222. IEEE, 1996.

[110] P. W. Wong. A public key watermark for image verification and authentication. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 455–459. IEEE, 1998.

[111] G. Woo, A. Lippman, and R. Raskar. Vrcodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 59–64. IEEE, 2012.

[112] 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.

[113] S. Zahid, M. Shahzad, S. A. Khayam, and M. Farooq. Keystroke-based user identification on smart phones. In *Recent Advances in Intrusion Detection*, pages 224–243. Springer, 2009.

[114] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu. Fingerprint attack against touch-enabled devices. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pages 57–68. ACM, 2012.

[115] J. Zhao and G. Mercier. Transactional video marking system, Mar. 14 2014. US Patent App. 14/214,366.

[116] N. Zheng, K. Bai, H. Huang, and H. Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pages 221–232. IEEE, 2014.