MarkIt: Privacy Markers for Protecting Visual Secrets

Nisarg Raval
Department of Computer Science
Duke University
nisarg@cs.duke.edu

Landon Cox
Department of Computer Science
Duke University
lpcox@cs.duke.edu

Animesh Srivastava
Department of Computer Science
Duke University
animeshs@cs.duke.edu

Ashwin Machanavajjhala
Department of Computer Science
Duke University
ashwin@cs.duke.edu

Kiron Lebeck
Department of Computer Science
Duke University
kkl11@cs.duke.edu

UbiComp '14, September 13 - 17 2014, Seattle, WA, USA

ACM 978-1-4503-3047-3/14/09$15.00.
http://dx.doi.org/10.1145/2638728.2641707

Abstract

The increasing popularity of wearable devices that continuously capture video, and the prevalence of third-party applications that utilize these feeds have resulted in a new threat to privacy. In many situations, sensitive objects/regions are maliciously (or accidentally) captured in a video frame by third-party applications. However, current solutions do not allow users to specify and enforce fine grained access control over video feeds.

In this paper, we describe *MarkIt*, a computer vision based privacy marker framework, that allows users to specify and enforce fine grained access control over video feeds. We present two example privacy marker systems – PrivateEye and WaveOff. We conclude with a discussion of the computer vision, privacy and systems challenges in building a comprehensive system for fine grained access control over video feeds.

Introduction

With increasing sophistication in understanding visual inputs and advancements in wearable gadgets, Augmented Reality (AR) has become a reality. Products like *Google Glass* and *Kinect* provide platforms where developers can build and publish AR applications. These platforms (e.g., *Google Play*) also provide an easy access to download and install such third-party applications. Many third-party applications capture camera input and perform some processing on it. For example, Word Lens[1] - a text translation app on *Google Glass* accesses a real-time video feed from *Glass*, detects text in the input, performs required translation and projects the translated video back onto the user's view. Strictly speaking, a translation app only needs textual content of the video, but instead it accesses the whole video even when no text is present! A malicious translation app can easily gather personal information about a user underneath the pretense of providing translation services. Even if the app is not malicious, it may inadvertently record personal information about the user. For example, a *Glass* user may accidentally share an embarrassing picture while using it.

In order to provide a personalized experience, many applications gather sensitive information for user profiling. With Internet of Things [11] and Life Logging camera [6], the extent to which such information can be gathered is unimaginable. The *always recording* feature of wearable devices can leak sensitive information like personal pictures, enterprise secrets, health information etc. Although, apps provide some level of functional access control, the user doesn't have fine grained control over what information is revealed and what information is kept private against these third-party apps.

Many privacy preserving techniques have been proposed in the context of AR technologies (see sidebar for a detailed description of related work). Most of these techniques involve a privacy framework with exclusive access to sensor input. All third-party applications request sensor data through the privacy framework which in turn allows these requests based on predefined access control policies. Current models for such a privacy framework can be classified into two categories. The "least privilege" approach allows an application to access only those information which are absolutely required for its functionality [7, 8]. Usually, an application requests an object of interest. Then, the privacy framework informs user about potential information leakage and prompts for her permission. The application is given access to the object only if user permits.

The "least privilege" approach works very well when the application needs are clearly defined and confined to specific objects (e.g., location, face etc.). However, this approach fails when application needs are either unspecified or very broad (e.g., background). For example, a navigation app like *iOnRoad*[2] may require general properties of an image to detect roads, lanes, parking spots, etc. To handle such scenarios, another approach "protecting secrets" is proposed in which a set of secrets (sensitive objects) are defined and are protected by the privacy framework [2, 3, 17]. Most of the proposed systems in this category protect only predefined sensitive objects. However, it is infeasible to construct an exhaustive list of potential sensitive objects. Also, the list of sensitive objects differ from person to person. To overcome these limitations, we propose the *MarkIt* privacy framework whereby users can specify arbitrary secrets.

The novelty of our framework is in the use of *privacy markers* that allow users to specify an arbitrary sensitive region or object in the video feed. We describe two example markers – special rectangle to protect secrets on a two dimensional surfaces (e.g., white board) and virtual bounding box drawn using hand gestures to protect three dimensional objects. The high level idea is to continuously look for privacy markers in an incoming video feed. Once,

Related Work Privacy in context of AR technologies has gained lot of attention among researchers in the past few years.
Jana *et al.* [7] proposed the Recognizer abstraction for fine grained access control in AR systems. The idea behind recognizer is to have least privilege access control for third-party applications. However, it is unclear how to preserve global properties (e.g., background) using recognizers.
Scanner Darkly [8] is a privacy framework for context-aware AR applications. Darkly is integrated with the popular computer vision library OpenCV and provides a secure interface to all third-party applications. Darkly provides access control at an image level as opposed to object level access control similar to recognizers.
Templeman *et al.* proposed PlaceAvoider [18] system which alerts user when an application tries to capture images of sensitive locations (e.g., bathroom).

[1] http://questvisual.com

[2] http://www.ionroad.com

Chaudhari *et al.* [2] used real-time audio distortion and visual blocking to protect privacy of subjects in a video. A similar problem was solved by Schiff *et al.* [17] using color based visual markers to detect and block faces. Enev *et al.* [3] proposed a novel approach of transforming raw sensor data such that the transformed output minimizes the exposure of user defined private attributes while maximally exposing application defined public attributes.

Recently, Roesner *et al.* [14] introduced new security and privacy challenges which need to be addressed in the context of AR technologies. They characterized these challenges along two axes - system scope and functionality.

Roesner *et al.* [15] argued that access control policies should be specified by the objects themselves in continuous sensing platforms. They also proposed a general framework for world-driven access control of sensor data.

the marker is detected, an object enclosed by the marker is considered secret and removed from the current as well as subsequent frames before passing it to any third-party application. The aim is to provide entire raw image except sensitive objects to third-party applications. Thus, it allows an application to extract as much information as possible from an image without disclosing sensitive objects/regions.

We make the following contribution in this paper. First, we enlist the desired properties of privacy markers. Then, using the marker approach, we propose two conceptual privacy systems suitable for two different use cases. Finally, we discuss computer vision, privacy and systems challenges in building a comprehensive privacy framework for fine grained access control on video feeds.

Privacy Markers

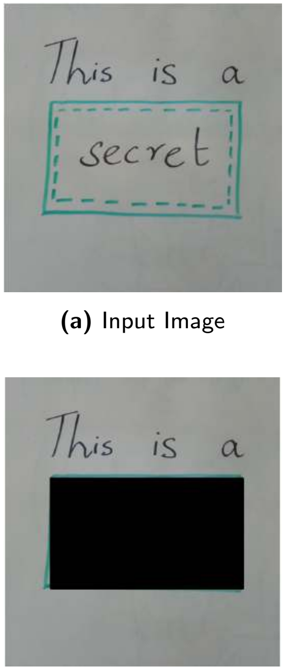
In this section, we briefly describe our ongoing work on *privacy markers* to specify and enforce privacy of visual secrets. We assume that the AR platform is trusted, but the applications that access the visual feed may be malicious. Our privacy specification and enforcement algorithm execute at the system level in the AR platform, while applications execute in user mode. This strategy allows us to enforce privacy before sending the video feeds to third-party applications.

The high level idea is to mark sensitive objects in real world using specialized *privacy markers* that can be efficiently detected by the AR platform in real-time. Once detected, the device blocks the sensitive object enclosed by the marker before passing the raw input image to third-party applications. These sensitive objects are protected throughout the subsequent frames. Thus, the system provides privacy guarantee against malicious third-party applications as well as accidental recordings. For an accurate and efficient implementation of the system, the privacy markers should possess the following properties.

- Uniquely identify and localize arbitrary objects of interest.
- Accurate and real-time detection by AR devices.
- Unambiguously encode a wide range of privacy policies.
- Easy to create/remove and cost effective (e.g., drawing by hand, digital tools or stickers).
- Invariant to occlusion, illumination and viewing angles.

Two possible options for privacy markers include Quick Response (QR) codes [1] and Radio Frequency Identification (RFID) tags [16]. However, they are not suitable for tagging visual secrets for the following reasons: Detecting a QR code and decoding relevant information depends upon uncontrolled aspects such as clarity of the QR code in the camera view of an adversary. The problem with RFID tags is that their performance is influenced by metallic objects around them [9]. Also, it requires pre-installed RFID readers on the devices.

Rather than depending on additional infrastructure, we show two examples of how simple human cues coupled with off the shelf computer vision algorithms can be used to specify and enforce privacy of visual secrets. We consider two popular scenarios of privacy breach and briefly discuss potential privacy markers for them. Although, these markers can encode various access control policies, in this paper we use the default policy of blocking the entire region enclosed by the marker. In future, we are planning to extend our markers to encode various access control policies.



(a) Input Image

(b) Output Image

Figure 1: Conceptual View of PrivateEye: The area enclosed by the marker (dotted rectangle within solid rectangle) is considered secret and blocked by the system.

PrivateEye
Consider a scenario where a sensitive product idea is being discussed on a white board in an enterprise environment. Imagine an employee walking in with a *Goggle Glass*. Even though, an employee may be trusted, its hard to trust third-party applications running on *Glass*. This may lead to malicious or accidental leaks of confidential information written on the white board. In order to prevent such leaks we need a mechanism such that no private information (white board content) can be accessed by third-party applications without prior authorization from the content owner. This mechanism also needs some means to define private information (area on a white board). We develop a simple privacy marker "dotted rectangle within a solid rectangle" to mark two dimensional area on a white board (or presentation slides). Any content enclosed within this marker is considered private by the system. This marker together with the algorithm for recognizing it result into our first solution *PrivateEye*.

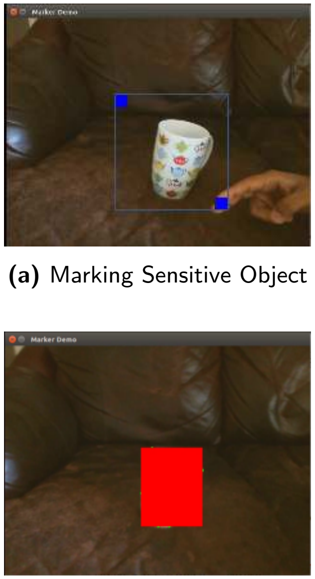
With PrivateEye, we take the first step towards building a system that can provide privacy to two dimensional regions against prying digital eyes. PrivateEye consists of two pieces: (1) a specification for marking a two-dimensional space as secret, and (2) software on a recording device for recognizing markings and obscuring visual secrets in real-time. We investigate the possibility of using computer vision algorithms to identify visually sensitive information and process the visual data in real-time. Figure 1 shows examples of how PrivateEye users can define a region containing visual secrets by combining solid and dotted lines. Depending on the medium, users can define secret regions by hand (e.g., on a white board) or use digital tools (e.g., within a presentation). When a sensitive visual information is in the camera view, the PrivateEye framework denies access of that region to any third-party applications.

WaveOff
The scope of secrets protected by PrivateEye markers is limited to two dimensional surfaces like white board, wall painting, presentation, etc. In many cases users may want to protect three dimensional objects. One such use case is to protect a keyboard while typing password. A malicious glassware can easily launch shoulder surfing attack just by recording the keyboard while user is typing her password. Blocking the entire view or turning off the *Glass* during password-entry may not be a good solution as the view might be useful for some applications. For example, one can think of a password manager which can superimpose special indicators over the correct keys to help user with the complex password [14].

We propose a novel approach of using hand gestures to mark real world sensitive objects in real-time. Using the proposed system *WaveOff*, user can draw a virtual bounding box in the air enclosing a sensitive object (e.g., keyboard). The system detects the virtual bounding box and marks the enclosed object as sensitive. Henceforth, that specific object will be blocked in all the consecutive frames. WaveOff only protects the sensitive object as long as it is in the field of view. Once, the object goes out of the camera view it is no longer protected. The conceptual view of WaveOff is outlined in Figure 2. WaveOff heavily uses computer vision algorithms for various tasks like detecting hand gestures, tracking sensitive objects, etc.

Challenges

In order to successfully implement proposed marker system, we need to address many challenges spanning across various domains. Broadly, we categorize them into three areas - computer vision, privacy and systems



(a) Marking Sensitive Object

(b) Blocking Marked Object

Figure 2: Conceptual View of WaveOff: User uses hand movements to draw virtual bounding box. The object enclosed by the bounding box is protected throughout all the subsequent frames.

challenges. In this section, we briefly outline those challenges and pointers to address them. Even though, our paper focuses on marker system, most of these issues are applicable to any system that protects visual secrets.

Computer Vision Challenges
The key component of the proposed marker system is identifying sensitive objects (or markers) in the input camera feed in real-time. This task involves three important computer vision techniques - object detection, recognition and tracking. The appearance of an object changes a lot with variation in illumination, poses, views, etc. This high variation in appearance makes both detection and recognition hard to solve. Even state of the art computer vision algorithms fail to achieve high accuracy [4].

With the help of privacy markers, we are able to reduce the harder problem of detecting an arbitrary object into much simpler problem of detecting a specific object (marker). Furthermore, one can design a marker (irrespective of the sensitive objects) which is easy to detect using current object detection algorithms. However, the same technique can not be applied in case of recognition because even with the fixed object, recognition in wild is an extremely hard problem. If we restrict our system to only protect sensitive objects as long as they are in the field of view, then we can avoid recognition. However, if we want the system to remember previously marked sensitive objects then we need to train our system to recognize objects.

Lastly, object tracking also faces various challenges like drifting, occlusion, etc. Some of the current vision algorithms attempt to solve these issues to certain extent. For example, Tracking-Learning-Detection (TLD) [10] simultaneously tracks the object, learns its appearance and detects it whenever it appears in the video.

Privacy Challenges
As explained in the previous section, the proposed system heavily uses various computer vision algorithms. Hence, the privacy guarantees of such a system highly depends on the accuracy of the corresponding computer vision algorithms. Although state of the art vision algorithms are nowhere near perfect, we argue that even with a 99% accurate computer vision algorithm, it is hard to provide an arbitrary privacy guarantee. To understand this, consider a simple use case of face detection. To preserve a person's identity, we blur all faces in the input video stream. More specifically, given a video stream, we run face detection algorithm on every frame and blur every face detected by the algorithm. Considering a video capturing rate of 25 fps, a person's face appearing for approximately 7 minutes, appears in more than 10000 consecutive frames. Even with a 99% accurate face detection algorithm, the probability that we miss a face in one of these 10000 frames is significantly large (assuming uniform failure rate). Revealing a face in even one frame reveals the identity of that person in the entire video. Although the above argument is ad-hoc as it doesn't take into account the fact that adjacent frames are correlated (face usually exist in consecutive frames), the argument that *achieving arbitrary privacy guarantee is infeasible even with a near perfect vision algorithm* is still valid.

In some cases, revealing a face in a single frame might be a breach of privacy; for example, when we want to keep the presence of a person secret. However, in other cases, revealing a face in a few frames might be acceptable. One example of such a scenario is *lip reading* from a video. Even if we reveal few non consecutive frames with faces, it is hard to identify spoken words. Intuitively, we can pose a question like “Which (or How many) frames need to be revealed in order to leak a specific secret?”. The answer to this question (span of frames or *extent*) can be used to characterize the secret. For example, if a person appears throughout the video, then revealing any frame reveals the existence of that person. Hence, the extent of the secret *existence* is very high. On the other hand, one needs to reveal lot of consecutive frames in order to infer spoken words. Thus, the extent of *lip reading* is relatively low. It is an interesting research direction to categorize visual secrets based on their spatial as well as temporal extent and analyze its relationship with privacy guarantee.

One of the distinctive features of visual secrets is that they are probabilistic as opposed to traditional resources which are deterministic in nature. Hence, we need a probabilistic access control mechanism to deal with visual secrets. Apart from their probabilistic nature, visual secrets are highly correlated. For example, if an object appears in a particular frame of a video, then with high probability the same object will exist in subsequent consecutive frames. Rastogi *et al.* [13] proposed an access control mechanism for probabilistic databases. However, we need a novel approach in designing access control mechanisms that account for both uncertainty and correlation among visual secrets.

Systems Challenges
Most of the AR devices available today are not equipped with high computing power[3]. On the other hand, many required vision algorithms are computationally intensive and sometimes need large storage (e.g., training) as well. Thus, the question is how to perform real-time computation over limited computational resources provided by AR devices? The most popular solution to this problem is *offloading*. The idea is to offload heavy computation on server/cloud infrastructure [5, 7, 8, 12]. However, this poses several other challenges like privacy/security issues on cloud, trade-off between computing time and network bandwidth, etc. One needs to carefully think about these issues in the context of available resources.

The need for real-time computation stem from the requirements of a typical AR application. Many AR applications require instantaneous access to the captured video feed (e.g., Word Lens). However, certain applications like photo sharing need not get instantaneous input from camera. It may be acceptable to introduce a minor delay to pre-process the video feed in order to remove sensitive objects. With more computing time one should get better privacy guarantee as the input image can be thoroughly checked for sensitive objects. There seems to exist a trade-off between computing time and privacy. We believe that the categorization of applications along the axes of delay-tolerance will lead to better understanding of this trade-off. This will in turn allow us to provide application specific privacy guarantees.

Conclusion

We proposed a novel approach of *privacy markers* that enables users to define arbitrary sensitive objects in the real world. We also presented two ongoing projects *PrivateEye* and *WaveOff* to explain how privacy markers can be used to protect visual secrets. We note that the proposed system is not a complete solution. We discussed a number of computer vision, privacy and systems challenges that need to be addressed to realize this vision, and hope this discussion fuel interesting interdisciplinary research in the future.

[3] Note, that future AR devices may possess high computing power but the corresponding algorithmic complexity will also increase.

References

[1] Belussi, L. F., and Hirata, N. S. Fast component-based qr code detection in arbitrarily acquired images. *J. Math. Imaging Vis.* (2013).

[2] Chaudhari, J., Cheung, S., and Venkatesh, M. Privacy protection for life-log video. In *SAFE* (2007).

[3] Enev, M., Jung, J., Bo, L., Ren, X., and Kohno, T. Sensorsift: Balancing sensor data privacy and utility in automated face understanding. ACSAC (2012).

[4] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[5] Ha, K., Chen, Z., Hu, W., Richter, W., Pillai, P., and Satyanarayanan, M. Towards wearable cognitive assistance. MobiSys (2014).

[6] Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N., and Wood, K. Sensecam: A retrospective memory aid. UbiComp (2006).

[7] Jana, S., Molnar, D., Moshchuk, A., Dunn, A., Livshits, B., Wang, H. J., and Ofek, E. Enabling Fine-Grained Permissions for Augmented Reality Applications With Recognizers. In *USENIX Security* (2013).

[8] Jana, S., Narayanan, A., and Shmatikov, V. A Scanner Darkly: Protecting User Privacy from Perceptual Applications. In *S & P* (2013).

[9] Juels, A. Rfid security and privacy: a research survey. *Selected Areas in Communications, IEEE Journal on* (2006).

[10] Kalal, Z., Mikolajczyk, K., and Matas, J. Tracking-learning-detection. *PAMI* (2012).

[11] Metz, R. More connected homes, more problems. *MIT Technology Review* (2013).

[12] Ra, M.-R., Sheth, A., Mummert, L., Pillai, P., Wetherall, D., and Govindan, R. Odessa: Enabling interactive perception applications on mobile devices. MobiSys (2011).

[13] Rastogi, V., Suciu, D., and Welbourne, E. Access control over uncertain data. *VLDB* (2008).

[14] Roesner, F., Kohno, T., and Molnar, D. Security and privacy for augmented reality systems. *Commun. ACM* (2014).

[15] Roesner, F., Molnar, D., Moshchuk, A., Kohno, T., and Wang, H. J. World-driven access control for continuous sensing. Tech. Rep. MSR-TR-2014-67, 2014.

[16] Sample, A., Macomber, C., Jiang, L.-T., and Smith, J. Optical localization of passive uhf rfid tags with integrated leds. In *RFID* (2012).

[17] Schiff, J., Meingast, M., Mulligan, D., Sastry, S., and Goldberg, K. Respectful cameras: detecting visual markers in real-time to address privacy concerns. In *IROS* (2007).

[18] Templeman, R., Korayem, M., Crandall, D., and Kapadia, A. PlaceAvoider: Steering first-person cameras away from sensitive spaces. In *NDSS* (2014).