Towards Privacy Protection of On-screen Information

Jingnan Wang
Shanghai Jiao Tong University
Minhang District, Shanghai, China

Wanghua Shi
s-whua@sjtu.edu.com
Shanghai Jiao Tong University
Minhang District, Shanghai, China

Figure 1. A demostration of shoulder surfing and our countermeasure

Abstract

People do not want their information from personal devices to be seen by shoulder surfers when they use the mobile devices. Utilizing the human vision and optical system properties, this paper proposes HideScreen to conceal the information shown on electronic screen from shoulder surfers. By dividing the screen into small and various grids, low-frequency components can be neutralized which means the obvious information mingles within the background when viewed from the outside of the designed range. HidenScreen has three protection schemes: HideText, HideImage and SelImage. We realize the effect of concealing information in text and image, just as shown in original paper.

CCS Concepts: • **Information** → Information Security.

Keywords: Hidescreen, Hidetext, Hideimage, Shoulder Surfers

ACM Reference Format:
Jingnan Wang and Wanghua Shi. 2021. Towards Privacy Protection of On-screen Information. In *Mobile Sensing and Computing*. SJTU, Shanghai, SH, CHINA, 4 pages.

1 Introduction

People use mobile phone or tablet PC everyday and everywhere. When in public area, it is easy to divulge personal and sensitive information shown on the screen because people nearby may peek at the screen. Few people take proper defensive actions when they beware of someone else's peeking at their screens.

There are some popular defensive measures against shoulder surfing. Attaching a privacy film on the device screen limits the visible range of screen to a certain viewing angle to hide the on-screen information and needs users to buy and attach a privacy film. IllusionPIN needs to appropriately tune the parameters for each concrete task to achieve the protection.

HideScreen needs not additional hardware. It can protect the sensitive information without compromising users' intended apps and is simple to implement and run on commodity devices while consuming little power.

The basic idea of HideScreen is injecting high spatial frequency components into privacy information and neutralizing the low-frequency components. Grid patterns are used to make the information can only be viewed within a designed range.

2 Model Assumption

1. Assume shoulder surfers use their eyes or smartphone cameras to acquire or comprehend the on-screen information.
2. Smartphones/tablets/laptops are assumed to be viewed by their users from a distance up to 24".
3. For surfers sitting behind the user, the distance between the device and the surfers will be the size of seat pitch, which is set to 28".
4. The surfers would not use a camera to video-record the user's device screen and process the video to extract sensitive information.

3 System Design

3.1 Overview

HideScreen has three protection schemes: HideText, HideImage and SelImage. All of these can protect information by viewing distance and angle. HideText targets at text protection and does not have loss. HideImage targets at image protection and has some content loss. SelImage aims at image protection and has no loss. Besides, it uses user interaction to provide protection.

3.2 Text Protection

3.2.1 Calculate the Grid Size. The formula to calculate grid size l is:

$$l = \lfloor \frac{d + 12"}{3333 \times l_p} \rfloor \times l_p \tag{1}$$

In the equation, d is the user's viewing distance, l_p is the pixel size.

3.2.2 Identify Text Boundaries. Treat the pixels that connect to the background pixed as boundary pixels. After obtaining the first boundary layer, treat it as background and find pixels next to the new background, which constitute the second boundary layer. Recursively do this operation until enough number of layers is found.

3.2.3 Replace Text Boundaries. Replace different boundary layer with different colors. Assume the number of boundary layers is three. Use the following formula to calculate the color of each layer:

$$H_L = \lfloor H_{BG} + \frac{(3 - L)(-1)^L}{10} H_{bright} \rfloor \tag{2}$$

In this equation, $L = 1, 2, 3$ are the indices of boundary layers, layer 1 is closest to the background, $H_{bright} = \#FFFFFF$.

3.2.4 Fill Background. Replace the background with a new color H_{BG}. $H_{BG} \approx H_{bright} \oplus H_{dark}$. $H_{dark} = \#000000$.

3.2.5 Some Tips.

1. We can encapsulate the hidetext function as a font file. Input the text then output the image showing the hiden text.
2. The bolder the font of text, the better the protection.
3. For simplicity, we can set $H_{BG} = (127, 127, 127)$ or directly convert the RGB image to grayscale image and set $H_{BG} = 127$ since the pictures used are achromatic.
4. It is not necessary to make sure the grid is a single color. For instance, when the grid size is a 2×2 pixel, it can be one color on the main diagonal and another color (e.g. H_{BG}) on the other diagonal.

3.3 Image Protection

HideImage causes some loss of information while providing protection. It utilizes different grids with same "average" color (H_{avg}) to represent a different brightness scale.

3.3.1 Transformation and Partition. Transform the image into grayscale and partition the greyscale image into layers of different color levels.

3.3.2 Replacement. Replace the identified color levels with grids of different bright–dark component combinations, which are determined as follows:

$$H_{avg} = \lfloor \frac{H_{white} + H_{dark}}{2} \rfloor \tag{3}$$

$$H_{dark,i} = \lfloor \frac{H_{avg} - H_{black}}{N_{level}} \times i + H_{dark} \rfloor \tag{4}$$

$$H_{avg} \approx H_{bright,i} \oplus H_{dark,i} \tag{5}$$

i represents the serial number of each level.

3.3.3 Illustration.

1. The parameter "number of levels", "grid size","background color" can be tuned to get better performance.
2. The mapping relation from original color value to color level can be adjusted.
3. The smaller size the picture, the clearer it is.

4 Proposed Solution and Evaluation

HideScreen [1] was proposed to protect our on-screen information from shoulder surfers. We implement HideText and HideImage.

4.1 Original HideText

Figure2 is the text protection example in [1] and figure 3 is the reproductive one. The essential modification is from the text boundary and the internal area in the text is not changed.

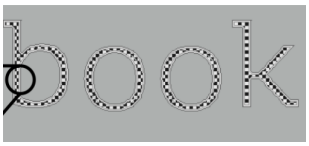

Figure 2. Original text protection example.

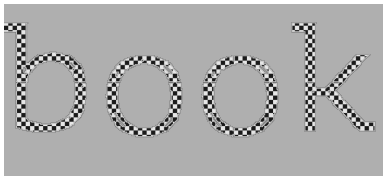

Figure 3. Reproductive text protection example.

4.2 Improved HideText

In our reproducing experiments, the original HideText method seems to be not good enough at hiding OSI from a far-away viewer. That is, a shoulder surfer may still see the sentences clearly through HideText. So we managed several attempts to improve the effect of text protection.

4.2.1 Switch to a new grid pattern. We noticed that the reason for the inadequate protection of original HideText is due to the high proportion the bright components. Human eyes are more sensitive to brightness and too much brightness can make the words obvious for shoulder surfers. So we changed the original gird pattern to ensure that dark components take the lead. Specifically, the dark components accounted for three quarters in the new grid pattern, as shown in figure 4. And we find that the new gird pattern actually provides better protection for on-screen text, as shown in figure 5.

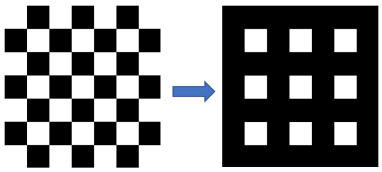

Figure 4. Switch to the new grid pattern

Figure 5. Improved HideText with new grid pattern

4.2.2 Change the border. Still, the HideText with new grid pattern should be further improved because it is obvious when being viewed from certain angles. Since the original boundary contains three different layers, we remove the bright layer from it and keep the other two layers. The result shows that such modification can reduce the visibility of shoulder surfers while maintaining the visibility of users, as shown in figure 6.

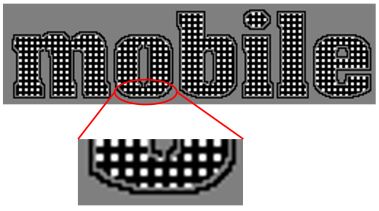

Figure 6. Improved HideText with new border

4.3 HideImage

The original and reproductive partition grid graphs are as follows:

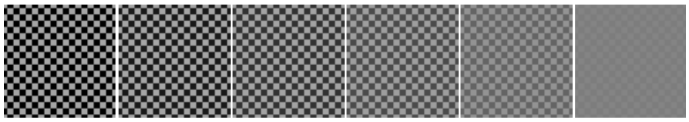

Figure 7. Original dark partition grids of six levels.

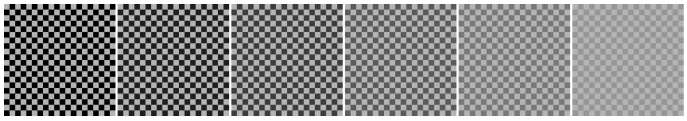

Figure 8. Reproductive dark partition grids of each levels.

The average color used is 180 and the grid size is 5 pixels. The grids of last two levels are brighter than the original grids, which may cause some deviation.

The white area in the picture 9 is the pixels that are within each original color level. Figure 10 shows the reproductive picture of figure 9.

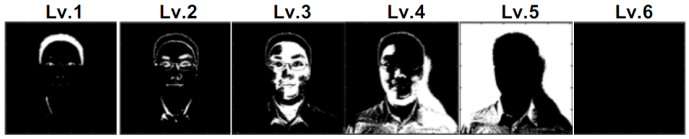

Figure 9. Original color level distribution.

Figure 10. Reproductive picture of color level distribution.

The first two levels perform well. The third and fourth pictures are not too bad. But there is a big gap between the last two pictures and the original pictures. Through large amount of experiments, we dirctly map the interval from one to two times of the level value to the respective level and then do replacement according to the map function. In this way, the last level is not used, and only 75 points belong to the fifth level in the example photo.

Figure 11 is the original photo. Figure 12 is the original protected image and figure 13 is the reproductive one.

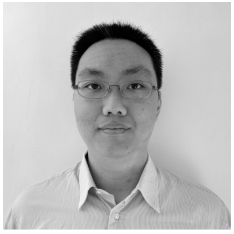

Figure 11. Original photo.

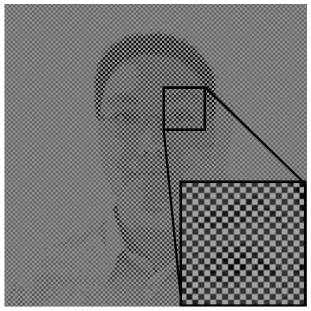

Figure 12. Original protected image.

Figure 13. Reproductive protected image.

However, all the experiments before use the value of the first position of grid to do the comparsion. It is more reasonable to use the average value in the grid to compare. In this way, we get more accurate results as follows($H_{avg} = 147$):

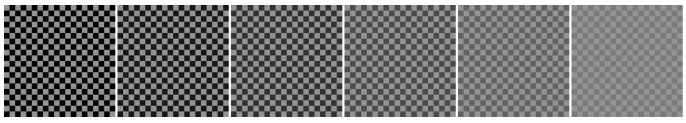

Figure 14. More accurate dark partition grids of each levels.

Figure 15. More accurate color level distribution.

Figure 16. More accurate protected image.

4.4 Problems

1. The calibration step is sophisticated and time-consuming, leading to a sense of user experience is poor.
2. HideScreen protect the OSI at the cost of loss of display, and one may feel uncomfortable after watching it for a long time.
3. The protective effect of HideImage is far from ideal as a close user may not see the hidden images clearly while a far-away shoulder surfer can.

5 Contribution

1. Wanghua Shi: implement the HideText and HideImage algorithms, and write report;
2. Jingnan Wang: implement the HideText and improve it, and make the slides.

Acknowledgments

To Mr. Chen, for the relaxed atmosphere in class and perseverance on teaching.

References

[1] Chun-Yu Chen, Bo-Yao Lin, Junding Wang, and Kang G Shin. 2019. Keep Others from Peeking at Your Mobile Device Screen!. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.