Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(6): 7458–7476.
DOI: 10.3934/mbe.2019374
Received: 30 March 2019
Accepted: 29 July 2019
Published: 16 August 2019

Research article

Exfiltrating data from an air-gapped system through a screen-camera covert channel

Longlong Li, Yuliang Lu*, Xuehu Yan and Dingwei Tan

National University of Defense Technology, No. 460 Huangshan Road, Hefei 230037, China

* **Correspondence:** publictiger@126.com.

Abstract: In recent years, many methods of exfiltrating information from air-gapped systems, including electromagnetic, thermal, acoustic and optical covert channels, have been proposed. However, as a typical optical channel, the screen-camera method has rarely been considered; it is less covert because it is visible to humans. In this paper, inspired by the rapid upgrades of cameras and monitors, we propose an air-gapped screen-camera covert channel with decreased perceptibility that is suitable for complex content. Our method exploits the characteristics of the human vision system (HVS) and embeds quick response (QR) codes containing sensitive data in the displayed frames. This slight modification of the frames cannot be sensed by the HVS but can be recorded by the cameras. Then, using certain image processing techniques, we reconstruct the QR codes to some degree and extract the secret data with a certain level of robustness due to the error correction capacity of QR codes. In the scenario to which our method applies, we assume that a program has been installed in the target system and has the authority to modify the frames without affecting the normal operations of valid users. Cameras, such as web cameras, surveillance cameras and smartphone cameras, can be receivers in our method. We illustrate the applicability of our method to frames with complex content using several different cover images. Experiments involving different angles between the screen and the camera were conducted to highlight the feasibility of our method with angles of 0°, 15° and 30°.

Keywords: covert channels, screen-camera communication, air-gapped, data exfiltration

1. Introduction

In the last decade, the development of the Internet has brought substantial convenience to humans; one can now communicate with another in less than one second, even when the other is thousands of miles away. However, the Internet also provides a way for hackers to attack any connected network. Despite considerable progress in defensive measures, e.g., intrusion detection and prevention systems (IDS/IPS), firewalls and antivirus programs, innovative attackers are still able to find ways to break into target systems. To protect information systems from the threats of anonymous attackers on the other side of the Internet, physical isolation, also called an air-gap, is used in nearly every top-level system. Cutting off direct contact between the target system and the Internet greatly increases the difficulty of in- or exfiltrating data.

Nevertheless, researchers remain motivated to propose novel methods of breaching air-gapped systems. Most of them belong involve four types of covert channel: Electromagnetic, acoustic, thermal and optical. Kuhn and Anderson [1] proposed the first method of transmitting data covertly using electromagnetic emanation in 1998. Guri et al. [2] designed AirHoppper, which leaks data through the electromagnetic radiation from computer screens. They also introduced a covert channel between a computer and a mobile phone using the electromagnetic radiation generated by the computer's memory bus in 2015 [3]. USBee [4], proposed in 2016, uses a covert channel based on a general USB device to emit modulated RF signals. Malley et al. [5] introduced a method for leaking data through inaudible sounds generated by computer speakers in 2014. Lee et al. [6] designed a covert channel between two speakers in 2015. Guri et al. tried to remove the device limitations and proposed Fansmitter in 2016 [7] and DiskFiltration in 2017 [8]. For thermal covert channels, Bitwhisper, which was proposed in 2015 [9], leaks data via heat radiation between two nearby computers. Mirsky et al. [10] introduced HVACKer in 2017; it uses an air-conditioning system to infiltrate data into an air-gapped system.

In the optical domain, LED indicators often act as signal emitters. Sepernitisky et al. [11] introduced a method of leaking data by modulating the frequency at which a monitor's power status indicator flashes. Lopes and Aranha [12] utilized a special storage device with an infrared LED to leak critical content. LED-it-GO [13] uses the hard drive LED as the optical signal, while xLED [14] uses status LEDs on the router. Keyboard LEDs emit optical signals in KLONC [15], leaking sensitive information to a nearby camera. Guri et al. [16] proposed a bidirectional communication prototype that uses surveillance cameras equipped with infrared LEDs and connected to an air-gapped network to emit modulated infrared light that can be received by the attackers outside and receive infrared signals from them covertly. However, objects between the LEDs and receivers may block the covert channel; these include the user's body, tables and chairs. As the main device displaying information to the user, the screen is fully or partly exposed and can serve as a signal source.

Jo et al. presented DisCo [17], which transmits messages by temporally and imperceptibly modulating the flashing frequency of the display. Hu et al. designed an optical sensor for detecting the flicker of an LED display and created a novel covert channel by controlling slight changes in the brightness level [18]. VisiSploit [19], proposed in 2016, embeds a QR code that contains a secret message on a bright/dark surface. It is made imperceptible by decreasing the contrast between the background and the image pixels, which relies on the characteristics of human vision. However, in practice, the bright or dark region on the screen is not very large when users are working on the computer, which limits the method.

The screen-camera method has been rarely considered because its visibility to humans makes it less covert. However, with the development of display and camera technologies, it has become increasingly practical to build screen-camera covert channels in air-gapped networks. First, there has been a significant improvement in display and camera resolution. Off-the-shelf screens are typically full HD or better, and most of cameras are at least 720p. Furthermore, surveillance cameras are installed in nearly every important place, including places with air-gapped networks. Last but not least, the security level of a surveillance system is usually lower than that of an air-gapped network, and some systems are even connected to the Internet. Besides, smartphones may be carried into the high security level place, and can be used to take photos and record videos. Guri et al. [19] has demonstrated the feasibility of screen-camera covert channels and inspired further research that utilizes the characteristics of the human vision system (HVS).

As we have noted, VisiSploit embeds secret images only on bright or black backgrounds, which seems to be a critical precondition. Therefore, in this paper, we propose a more general method that is applicable to most general background images, such as computer wallpapers, application windows, and videos. According to a spectrum test of LCD displays, the wavelengths of light generated ranges approximately from 400 nm to 750 nm, which in the range visible to humans. Consequently, it is inevitable that covert information transmitted via a single display frame can be received by both cameras and humans. Therefore, the key issue is to embed the secret in a way that makes it imperceptible to the HVS. Using the concept of informed steganography, our method first chooses two sequential frames to be displayed on the screen. Because the screen refreshes 60 times per second, it is easy to find two similar frames. Then, we leave the first unchanged and embed a secret QR image into the second by modifying its pixels slightly in a way that does not arouse the suspicions of valid users. Finally, the secret QR code can be reconstructed from the videos acquired by surveillance cameras or other nearby cameras.

Our work makes three contributions:

a. We proposed a method to exfiltrate data from an air-gapped system through a screen-camera channel. Our method is covert and does not attract suspicion even when the users are working on the computer.

b. Compared to VisiSploit [19], our method can embed data into more complex color images, not only images with bright or black backgrounds.

c. We conducted experiments to demonstrate the practicality of our method, especially for different camera-screen angles.

The rest of this paper is organized as follows: In section 2, we introduce the HVS, screen-camera communication and QR codes. Section 3 describes our method. Then, we describe experiments to test our method in section 4. Countermeasures are presented in section 5, and we conclude our work in section 6.

2. Preliminary

2.1. Human vision system

The human vision system is one of the most sophisticated optical systems in the world. Visible light passes through the cornea, iris, pupil, lens and vitreous, and then is received by rods and cones in the retina [20]. Rod cells are mainly distributed around the edge of the retina and are extremely sensitive to light and shade. These cells are mainly responsible for vision in low light levels but have difficulty distinguishing colors. Cone cells are mainly distributed in the central part of the retina. According to experimental results, there are three types of cone cells in human eyes: Cells that sense red, green and blue light. They are active at higher light levels [21].

The HVS is limited in its ability to identify changes in luminance. The visual increment threshold, also known as the just-noticeable difference or the luminance difference threshold, is defined as the brightness ΔI that just can be discerned from a reference field of intensity I. As shown in Figure 1, the visual increment threshold is a function of the reference intensity. At low light intensities, the threshold is constant; then, as the intensity increases, it enters the De Vries-Rose region with a slope of 1/2. The following region is known as the Weber region and is characterized by a logarithmic curve with unit slope. Finally, the saturation region is entered, in which the light is so bright that it is difficult for observers to notice a difference [22, 23].

Flicker fusion is another special characteristic of human eyes. When light flickers at a sufficiently high frequency, which is called the critical flicker frequency (CCF), time-variant fluctuations of light intensity are not perceptible by human eyes [24]. The CCF is affected by many factors, including the luminance, flickering area, and waveform. Consequently, the 60 Hz flickering presented by a monitor is above the CCF, and we do not notice the flickers of the background light source.

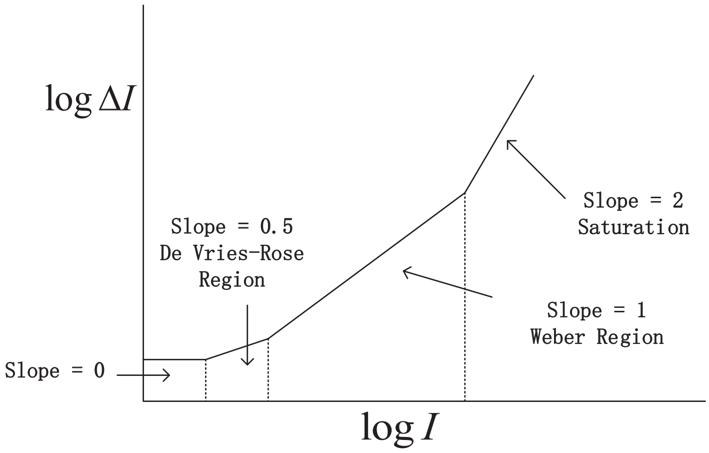


Figure 1. Linear approximation of increment threshold log ΔI as a function of the reference intensity log I.

2.2. Screen-camera communication

Alongside advances in display and camera technology, screen-camera communication has attracted a great deal of interest in recent years. Some studies have focused on proposing new two-dimensional coding methods for improving capacity, synchronization, transmission speed and robustness [25–27]. Imperceptibility is also an important characteristic that needs to be considered. HiLight [28] modifies the alpha channel of images to encode data covertly, while InFrame [29] and InFrame++ [30] display encoded images at 120 FPS to present normal frames to human vision.

Although screen-camera communication methods have been studied in depth, it is still not easy to directly apply them to air-gapped covert channels. First, quite a few methods cannot transmit data unobtrusively because they are based on the idea of constructing 2D barcodes with arresting patterns. Second, although some methods provide imperceptibility, they require displays with high refresh rates, which are not available in most air-gapped systems. Furthermore, photometric and geometric distortion is more severe than general screen-camera communication scenes because it is not convenient to freely adjust the relative positions and angles of screens and cameras in air-gapped systems.

2.3. QR codes

The quick response (QR) code was first designed in 1994 and became popular around the world due to its fast readability and high capacity. It consists of many black squares arranged on a white square grid background. Each square is defined as a module, and the number of modules is determined by the version of the QR code, which ranges from 1 to 40. A version 1 QR code consists of 21 modules per row and column. Then, the number of modules in each row and column increases by 4 with each version. The structure of a version 7 QR code is shown in Figure 2. Every QR code includes three conspicuous position patterns that are used to detect the QR code and determine its orientation. Alignment and timing patterns are also important to ensure that the recognized modules are in the right positions, which is especially crucial in a large QR code with many modules. The version and format information are encoded using a BCH code and appear twice to provide redundancy.

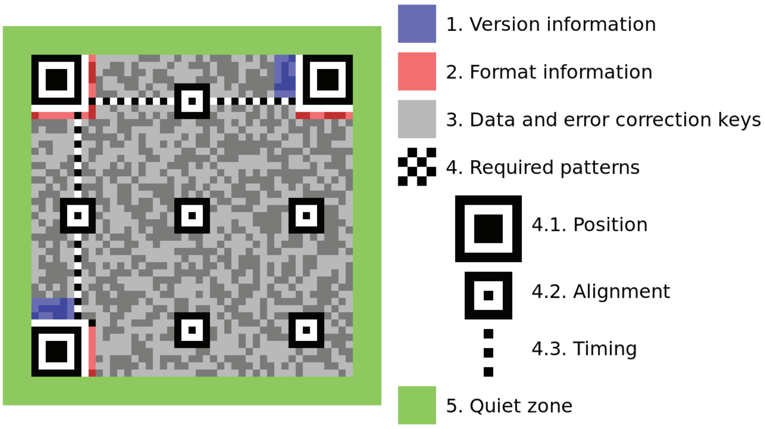


Figure 2. Structure of a QR code (version 7).

Another attractive advantage of the QR code is its error correction ability, which includes four levels (L = 7%, M = 15%, Q = 25%, H = 30%). When a QR code is generated, messages are first encoded to bit streams, and every 8 bits form a codeword. According to the vision of the QR code, the data codewords are divided into several blocks, which are the basic units for calculating error correction codewords. Therefore, the error correction is separated, and when the overall error rate is below the error correction capacity, the QR code can be recovered. Since the codewords for different blocks are sequentially arranged, partial damage to a QR code has a nearly equal effect on each block.

3. Method

Our method is shown in Figure 3. It consists of two parts, image concealment and reconstruction. The first is used to encode information into a QR code and then to embed the QR code into the displayed frames. The second involves reconstructing the QR code from the videos recorded by cameras.

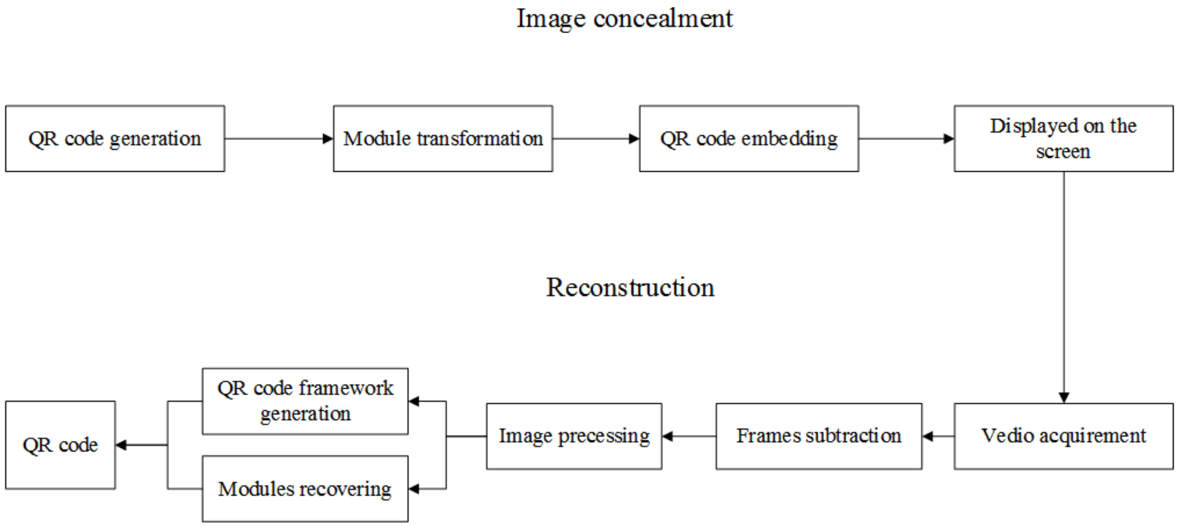


Figure 3. Illustration of our method.

3.1. *Scenario*

For our covert channels, infecting a computer in an air-gapped system is one of the basic requirements. We assume that a program has been installed on the computer and that it can search and collect sensitive information (e.g., passwords or secret text), which can be encoded as several QR codes. Then, the program embeds these QR codes into the normal frames covertly, without attracting any suspicion. Cameras are also needed as receivers of secret frames. We believe that surveillance cameras are a practical choice. Currently, surveillance systems are widely used, especially in top-secret organizations, to make sure no suspicious people slip in and staff behave as required. However, their level of security is usually lower than that of an air-gapped system, and some are even connect to the Internet. Therefore, invading a surveillance systems is more practical and easier, as has been shown in recent studies [31]. Other cameras that have access to the screens in a air-gapped network can also become receivers by recording the secret frames, e.g. web cameras and smartphone cameras. When the frames are acquired, the secret QR codes can be reconstructed using our method. QR codes make our work robust, but it must be admitted that encoded information may not be revealed if the reconstructed QR codes contain errors that exceed their capacity for error correction.

3.2. *Image concealment*

As previously mentioned, the HVS is limited in its ability to distinguish between two light sources with similar luminance. Therefore, we first choose two sequential frames to be displayed on the screen. Then, the former remains unchanged, and a secret QR image is embedded into the latter by modifying its pixels slightly in a way that does not arouse the suspicions of valid users.

3.2.1. QR code generation

QR codes have several parameters that need to be confirmed in the generating phase, including the version, module size, error correction level and mask pattern. The version and module size are determined by the size of the screen and the minimal unit that can be discriminated by the camera. The error correction level is set by default to H, the highest level with 30% error correction capacity. The noise between the screen and the camera is so substantial that we rely on error correction to provide some robustness, even though it decreases the data capacity of every single QR code. The mask pattern is also set to a default. On the one hand, the mask pattern is used for balancing the proportion of white and black modules, which is not very important in our method. It is acceptable even if the default mask pattern is not the optimal one. On the other hand, with the default error correction level and mask pattern, the QR code format can be confirmed. Finally, a binary QR code of size $N \times N$ is generated.

3.2.2. Module transformation

In general, the modules of QR codes are black and white squares. When a QR code is embedded into a cover image, we only change the values of the pixels corresponding to the black modules in the QR code. To decrease the number of modified pixels, we change the shapes of the black modules. As shown in Figure 4, the black modules are transformed to smaller shapes that are approximately circular. The original module size is 10×10 pixels, and after transformation, it is 45 pixels. This is a simple operation, but over 50% of the pixels in each black module require no modifications, which contributes a great deal to the code's imperceptibility.

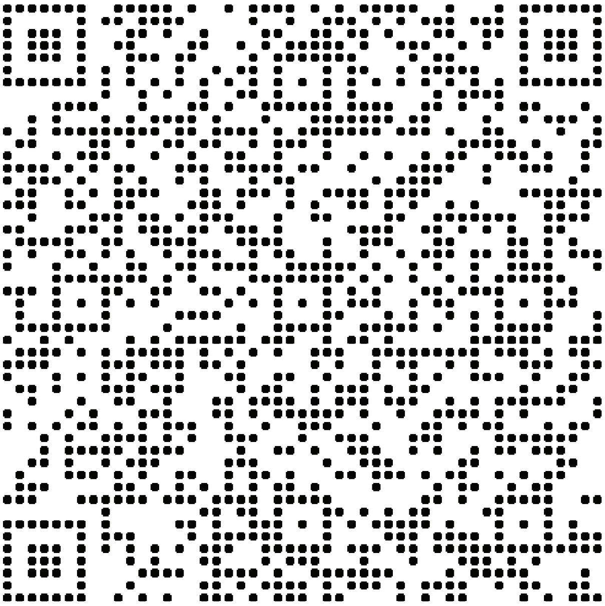


Figure 4. QR code with transformed modules.

3.2.3. QR code embedding

In most off-the-shelf displays, the default frame refresh rate is 60 Hz, which means that 60 frames are drawn on the screen each second. It is easy to find two neighboring frames that are nearly identical. We assume that two frames are selected and denote them by f_1 and f_2. A region R of size $N \times N$ is at the same position as f_1 and f_2 and serves as the cover image i_c into which QR code s is embedded. Color images have three channels: Red, green and blue. For applicability to most images, we perform the same operations on all three channels.

The operation applied to each channel is difference addition. When pixel $i_c(i, j)$ $(0 \leq i, j < N)$ of cover image i_c, if $s(i, j)$ $(0 \leq i, j < N)$ in the QR code equals 0, we apply difference addition. The differential d is based on the trade-off between imperceptibility and the reconstruction rate. If $i_c(i, j)+d > 255$, which means that it is out of the range of pixel values, the modified $i_s(i, j) = i_c(i, j)-d$. Otherwise, $i_s(i, j) = i_c(i, j) + d$. When $s(i, j)$ does not equal 0, $i_s(i, j) = i_c(i, j)$.

By traversing every pixel of i_c, we obtain the image i_s that includes an embedded secret QR code s. Then, we replace the region R of f_2 with i_s, generating a modified frame $f_2^{'}$. f_1 and $f_2^{'}$ are displayed on the screen, where they work as an information-transmitting pair. There is a flicker due to the abnormal difference between f_1 and $f_2^{'}$, which has an approximate frequency of 60 Hz. However, considering its amplitude and the number of modified pixels, the flicker is unlikely to attract much attention. Regardless, the interval between modified frames must be long enough such that the flickering can be ignored.

3.3. Reconstruction

Once a video of a monitor has been acquired, we split it into frames and apply subtraction to neighboring pairs, storing absolute values as differential images. From those images, it is easy to identify the embedded QR code; see, e.g., Figure 5, which has been processed to become visible. To remove the geometric distortion, a perspective transformation is needed. Then, we separate the blurry QR code and infer its version by counting the modules in a line. As we have discussed in the QR code generation phase, several parameters are set to default values, including the error correction level and the mask pattern. Therefore, we are able to reconstruct the framework of the QR code first, as shown in Figure 6, in case the version and format information are recovered incorrectly, which results in a failure to decode the QR code. The colored raw QR code is transformed to a grayscale image; each module is recovered according to the local gray distribution. When the number of pixels in a module with values above a certain threshold is large enough, which means pixels have been modified there, the module is most likely to be black. In the unchanged parts, the pixels appear random due to noise from many sources. Finally, we combine the framework and the rest of the recovered modules and obtain a QR code with some errors. The ultimate result is a standard QR code, and if the number of errors is below below the error correction capacity, the QR code can be decoded. For example, Figure 7 is a reconstructed QR code with a 21.07 % codeword error rate that remains readable.

4. Test and evaluation

4.1. Experimental setting

We use a 23-inch LEN T2324C LCD monitor with a resolution of 1920 × 1080. An industrial camera(MER-131-210-U3C) is used to take photographs; its resolution is 1280 × 1024. We choose 18 color images of size 512 × 512 as cover images. They are shown in Figure 8. We display a cover image 59 times and its modified partner once per second to simulate embedding a version 8 QR code in frames with a rather low frequency. Experiments are conducted indoors during working hours, and both the screen and the camera are exposed to ambient light, including natural light and LEDs. We test different screen-camera angles, namely, $0°, 15°, 30°$ and $45°$, with the grayscale images, while the

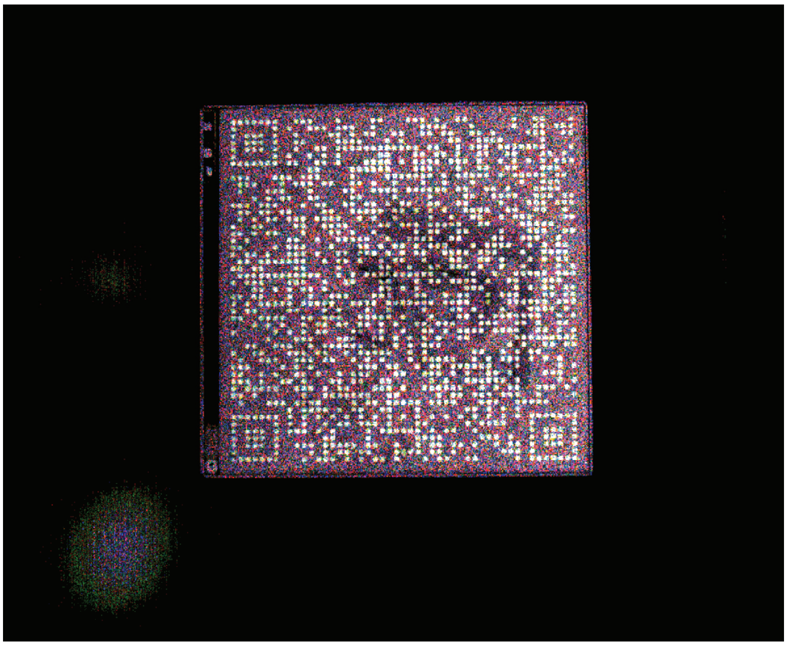


Figure 5. The differential image of two frames.

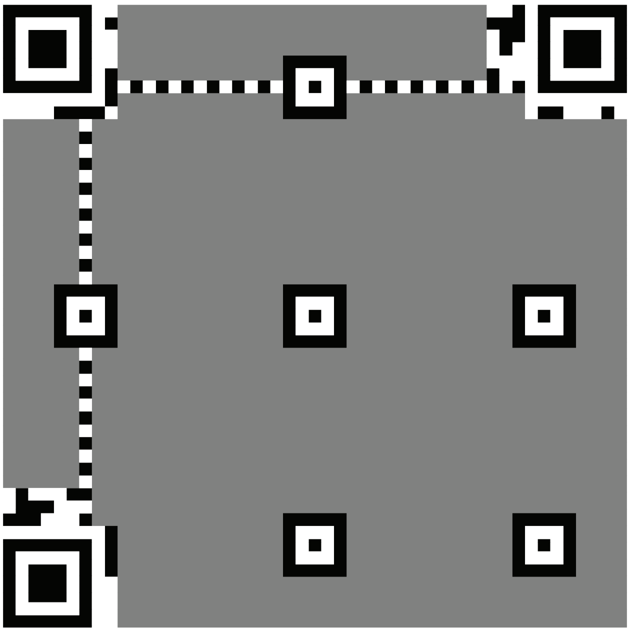


Figure 6. The framework of a version 8 QR code.

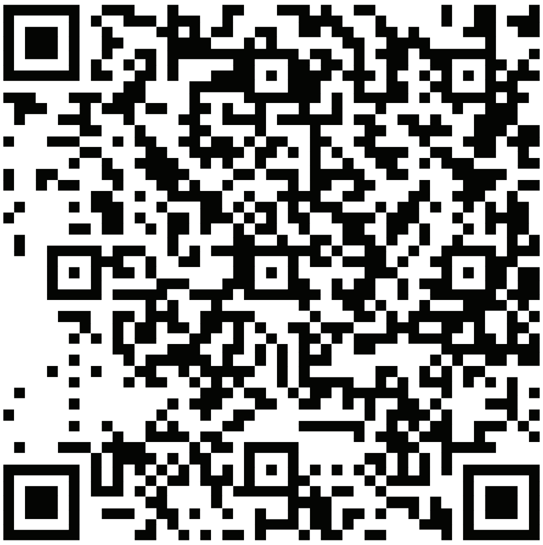


Figure 7. A reconstructed QR code.

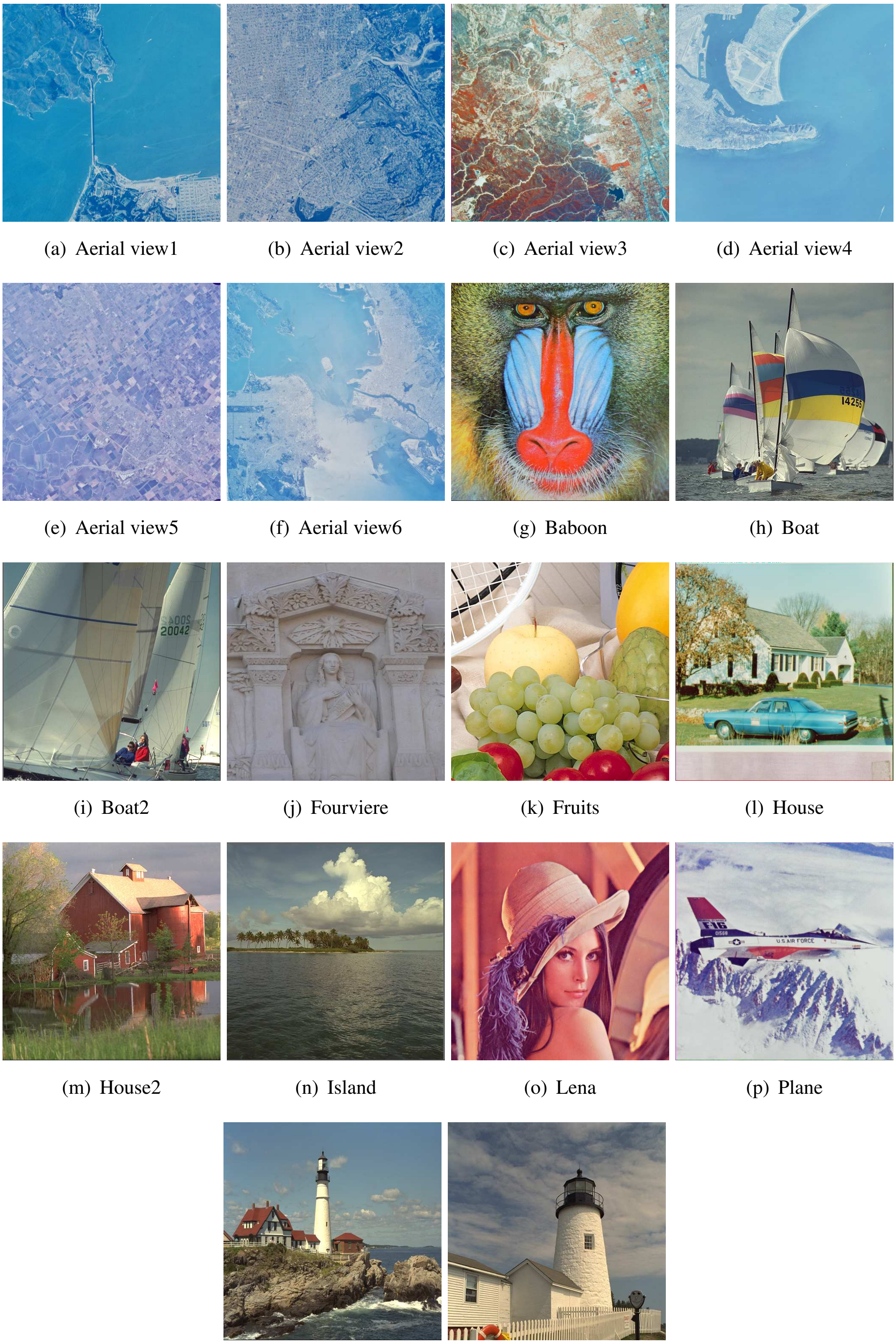
colored images are only tested at 0°. In order to illustrate the feasibility of our method, we then conduct experiments with a web camera and a mobile phone camera. The web camera is Logitech c310; its resolution is 1280 × 720. The mobile phone is MEIZU 16th plus and works wtih resolution of 1920 × 1080.

To evaluate our method, we choose several indicators of two features. For visual quality, the peak signal to noise ratio (PSNR) and the structural similarity index (SSIM) are often used to estimate the difference between two images. The SSIM considers the characteristics of the HVS by representing a human perception in a way. Generally, when the PSNR of two images is above 30 dB, we think that the difference is small enough. The SSIM ranges from 0 to 1, with values closer to 1 representing higher similarity.

The reconstruction results are evaluated using the module error rate, the codeword error rate and the decodability. The module error rate is defined as the ratio of the number of incorrectly recovered modules to the total number of modules. Since error correction in QR codes is based on a codeword formed by eight modules, the module error rate cannot apparently indicate whether the QR code is decodable. The codeword error rate calculates the error ratio of the codewords but does not consider the effect of the block partitioning. Nevertheless, it can be compared with the error correction level. Finally, a decoder plays the role of judge to provide a definitive answer. Although decodability is the most important indicator of the success of the reconstruction process, the module error rate (MER) and the codeword error rate (CWER) are useful ways of identifying the effects of different experimental objects.

4.2. Results

The differential d has a crucial influence on the imperceptibility and the reconstruction rate. Larger values of d can lead to higher reconstruction rates, but they increase the risk of attracting attention from users. We test different values to seek a balance, and finally we set d to 10. The PNSRs and SSIMs of the cover and modified images are shown in Table 1. The PSNRs are similar, and the



(a) Aerial view1 (b) Aerial view2 (c) Aerial view3 (d) Aerial view4

(e) Aerial view5 (f) Aerial view6 (g) Baboon (h) Boat

(i) Boat2 (j) Fourviere (k) Fruits (l) House

(m) House2 (n) Island (o) Lena (p) Plane

(q) Scenery (r) Tower

Figure 8. Cover images.

lowest is 34.85 dB, which means that the modified images are of acceptable quality. In contrast, the SSIMs are sparsely distributed. We find that images with more complex textures have larger SSIMs than smoothed images. Consequently, embedding QR codes into the regions with complex textures improves the imperceptibility, which is consistent with our subjective visual perception during the experiments.

Table 1. PSNRs and SSIMs of the cover and modified images.

Images	PSNR	SSIM
aerialview1	34.85	0.8909
aerialview2	34.85	0.9795
aerialview3	34.86	0.9780
aerialview4	34.68	0.8736
aerialview5	34.85	0.9561
aerialview6	34.85	0.9074
baboon	34.94	0.9762
boat	34.87	0.9074
boat2	34.87	0.9009
fourviere	34.85	0.9100
fruits	35.40	0.9306
house	34.85	0.9342
house2	35.05	0.9405
island	34.87	0.9211
lena	34.90	0.9275
plane	34.86	0.9108
scenery	34.90	0.9185
tower	34.99	0.9072
Average	34.91	0.9230

We use the industrial camera to photograph the screen and identify the pairs with embedded QR codes. After the reconstruction process, we obtain a recovered QR code and records of the module and codeword error rates, which are shown in Table 2. All the recovered QR codes can be decoded successfully. The results illustrate the validity of our method.

4.2.1. Influence of angles between the screen and camera

Normally, there is an angle between the screen and the camera, especially when the camera is a surveillance camera. We test our method with different angles, namely, $0°, 15°, 30°$ and $45°$, and calculate the codeword error rate of the gray cover images in Table 3 and Figure 9. Interestingly, as the angle increases, the codeword error rate first decreases for some groups, and most group produces

Table 2. Reconstruction of embedded QR codes at 0° with an industrial camera.

Images	MER(%)	CWER	Decodablity
aerialview1	0.42	4.95	Yes
aerialview2	0.42	3.3	Yes
aerialview3	1.26	9.91	Yes
aerialview4	1.69	10.33	Yes
aerialview5	0.53	5.37	Yes
aerialview6	0.3	3.3	Yes
baboon	2.03	12.39	Yes
boat	4.07	21.9	Yes
boat2	1.84	11.15	Yes
fourviere	0.23	2.06	Yes
fruits	4.88	21.9	Yes
house	2.34	18.18	Yes
house2	1.38	7.02	Yes
island	2.76	9.91	Yes
lena	3.03	19	Yes
plane	0.88	7.02	Yes
scenery	4.57	23.55	Yes
tower	3.11	13.63	Yes

the best result at 30°. We believe moire fringes are the main reason for the better results at 30°. When photographing screens, we see odd stripes and patterns, which are generated by interference between the pixel grids and the light sensors of the camera. The effects can be weakened by adjusting the angle and distance. However, 45° is too large an angle for an effective photograph because of the severity of the reflected light. Nevertheless, some groups can still be correctly decoded.

4.2.2. Test with different cameras

We conduct experiments using a web camera (Logitech c310) and smartphone camera (MEIZU 16th plus). Since the two cameras have lower performance than the industrial camera, we reduce the version of the QR code to three. Therefore, every module contains 15 × 15 pixels. We continuously display the images at a frequency of about 30 Hz, which means every 30 frames contain one secret frame. Switch cover images every 5 second. simultaneously, we respectively record a video using the two cameras. After splitting the video and carrying on the reconstruction process, we get the results, shown in Figure 4. The experiment set using web camera has 8 successful groups, and another set using smartphone camera has 11 successful groups. When analyzing the split frames, we find the interference of moire fringes is various in different frames, and some cover image groups show severer moire fringes. Besides, the web camera and the smartphone camera would adaptively adjust their parameters according to the light. Therefore, as for some parts in cover images, the modified pixel values may become unsensible by the cameras, leading to high codeword error rate. However, we can choose the images with low codeword error rate as the covers. In short, our method is practical with

Table 3. Codeword error rates for different screen-camera angles with an industrial camera.

Images	0 degree	15 degree	30 degree	45 degree
aerialview1	4.95	2.47	0	17.35
aerialview2	3.3	0	2.47	25.2
aerialview3	9.91	2.06	0.41	39.25
aerialview4	10.33	3.3	1.23	43.8
aerialview5	5.37	2.06	0	35.53
aerialview6	3.3	0	0.41	42.14
baboon	12.39	9.09	8.67	54.95
boat	21.9	11.15	8.67	37.19
boat2	11.15	20.66	4.13	30.99
fourviere	2.06	0.41	0	30.99
fruits	21.9	22.72	26.44	71.48
house	18.18	12.39	10.33	20.24
house2	7.02	3.71	2.47	26.03
island	9.91	7.85	5.37	19
lena	19	18.18	21.07	54.95
plane	7.02	7.85	7.43	54.13
scenery	23.55	13.63	9.09	29.68
tower	19.42	13.63	8.67	24.79

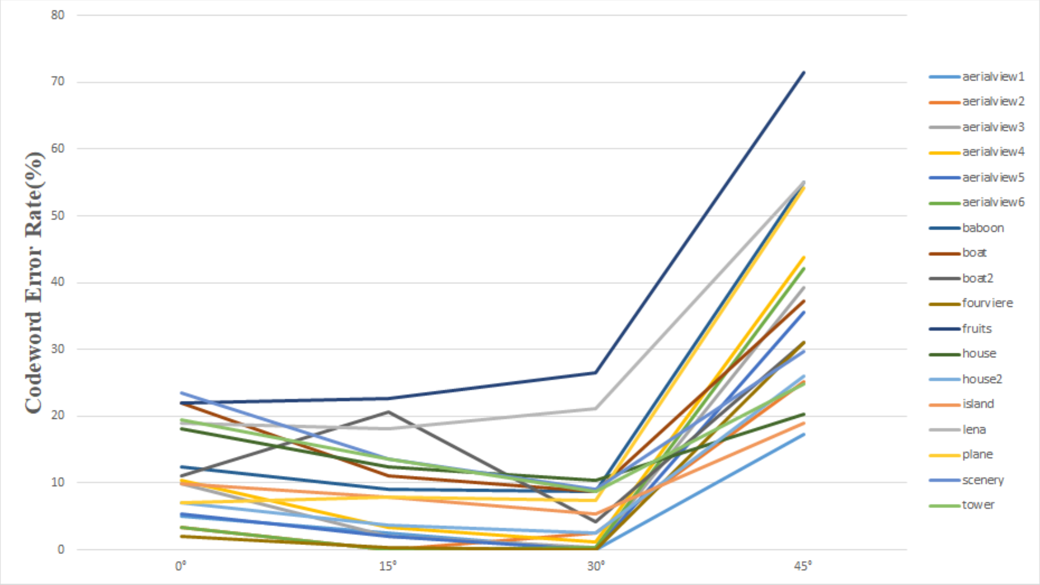


Figure 9. Codeword error rates for different screen-camera angles.

common cameras.

Table 4. Codeword error rates for different cameras.

Images	industrial camera		Web cammera		Smartphone camera	
	CWER(%)	Decodable	CWER(%)	Decodable	CWER(%)	Decodable
aerialview1	4.95	Yes	20	Yes	15.36	Yes
aerialview2	3.3	Yes	32.85	No	13.5	Yes
aerialview3	9.91	Yes	40.34	No	32.35	No
aerialview4	10.33	Yes	25.71	Yes	24.82	Yes
aerialview5	5.37	Yes	21.42	Yes	27.92	Yes
aerialview6	3.3	Yes	5.71	Yes	14.86	Yes
baboon	12.39	Yes	58.57	No	45.23	No
boat	21.9	Yes	37.14	No	37.87	No
boat2	11.15	Yes	17.14	Yes	24.52	Yes
fourviere	2.06	Yes	17.14	Yes	13.75	Yes
fruits	21.9	Yes	20	Yes	38.23	No
house	18.18	Yes	51.42	No	34.19	No
house2	7.02	Yes	32.85	No	28.76	No
island	9.91	Yes	31.42	No	36.84	No
lena	19	Yes	37.14	No	22.59	Yes
plane	7.02	Yes	35.71	No	21.4	Yes
scenery	23.55	Yes	32.85	No	23.96	Yes
tower	19.42	Yes	27.14	Yes	8.67	Yes

4.3. Comparison

Compared to Guri et al.'s method [19], our method has advantages in terms of perceptibility and applicability to complex color images. From the modified image quality test, we find that texture in the cover images improves the SSIM, which leads to better covert performance. With bright or black images, there is no texture; therefore, it is easier for them to arouse suspicion under the same conditions. Furthermore, bright and black images can be used in our method, and only one frame is needed.

5. Countermeasures

Countermeasures may include more strict management of the accessibility of air-gapped computers. Sensitive equipment can only be accessed by highly authorized staff and should be kept away from computers and mobile devices that can be connected to the Internet. No cameras should be carried within the perimeter of an air-gapped system. However, surveillance cameras are required in most places with high security, and it is very important to protect surveillance systems from attacks. Another countermeasure is to scan and detect the displayed images frequently, which seems to be ineffective without knowledge of the specific image-embedding method. A more effective countermeasure is the use of privacy screen films, which only give the user in front of the screen a clear view; others see a darkened image. Their primary application is preventing "shoulder surfing" in public places. The drawback of such films is that they decrease the brightness of the screen and make users uncomfortable. Therefore, these films are not widely used. Furthermore, within a 30° angle, the screen's contents are still recognizable. Some privacy screen films only prevent shoulder surfing from the left and right sides. The view from above is visible.

6. Conclusion

In this paper, we present an optical air-gapped covert channel based on screen-camera communication. QR codes are chosen to encode secret information; they provide robustness due to their capacity for error correction. Then, they are embedded into frames to be displayed on the screen, which are captured by cameras, and the QR codes are reconstructed. Instead of hiding information in only one frame, we involve two neighboring frames, as in the informed steganography methods. In this way, the sight modification of one frame can be recovered by comparing it to the other one. We test our method on three color images and six gray images, and the results show that the method works. Furthermore, we conduct experiments at different angles, and at 30°, we obtain the best reconstruction results. Compared to Guri et al.'s method [19], our method can embed QR codes in more complex images; it is not limited to images with bright or black backgrounds. Future work may include deeper research on the HVS and attempts to use more of its characteristics to improve the imperceptibility. Another direction is to consider the improvements of high-FPS monitors. It is challenging to use cameras with low frame rates, such as surveillance cameras, to capture effective photos and videos from which the embedded data can be recovered.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China (grant number 61602491).

Conflict of interest

The authors declare no conflicts of interest.

References

1. M. G. Kuhn and R. J. Anderson, *Soft tempest: Hidden data transmission using electromagnetic emanations,* International Workshop on Information Hiding, 1998, 124–142. Available from: https://link.springer.com/chapter/10.1007/3-540-49380-8_10.

2. M. Guri, G. Kedma, A. Kachlon, et al., *Air hopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies*, Proceedings of the 9th IEEE International Conference on Malicious and Unwanted Software: The Americas (MALWARE), 2014, 58–67. Available from: https://ieeexplore.ieee.org/abstract/document/6999418/.

3. M. Guri, A. Kachlon, O. Hasson, et al., *GSMem: Data exfiltration from air-gapped computers over GSM frequencies,* 24th USENIX Security Symposium (USENIX Security 15), 2015, 849–864. Available from: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/guri.

4. M. Guri, M. Monitz and Y. Elovici, *USBee: Air-gap covert-channel via electromagnetic emission from USB*, 2016 14th Annual Conference on Privacy, Security and Trust (PST), 2016, 264–268. Available from: https://ieeexplore.ieee.org/abstract/document/7906972.

5. S. O'Malley and K.-K. Choo, *Bridging the air gap: Inaudible data exfiltration by insiders*, 20th Americas Conference on Information Systems (AMCIS 2014), 2014. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2431593.

6. E. Lee, H. Kim and W. Y. Ji, *Various threat models to circumvent air-gapped systems for preventing network attack,* International workshop on information security applications, 2015. Available from: https://link.springer.com/chapter/10.1007/978-3-319-31875-2_16citeas.

7. M. Guri, Y. Solewicz, A. Daidakulov, et al., Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers, *arXiv preprint arXiv,* (2016).

8. M. Guri, Y. A. Solewicz, A. Daidakulov, et al., Diskfiltration: Data exfiltration from speakerless air-gapped computers via covert hard drive noise, 98–115. *arXiv preprint arXiv: 1608.03431,* (2016).

9. M. Guri, M. Monitz, Y. Mirski, et al., *Bitwhisper: Covert signaling channel between air-gapped computers using thermal manipulations*, 2015 IEEE 28th Computer Security Foundations Symposium, 2015. Available from: https://ieeexplore.ieee.org/abstract/document/7243739.

10. Y. Mirsky, M. Guri and Y. Elovici, Hvacker: Bridging the air-gap by manipulating the environment temperature, *Magdeburger J. zur Sicherheitsforschung,* **14** (2017), 815–829.

11. V. Sepetnitsky, M. Guri and Y. Elovici, *Exfiltration of information from air-gapped machines using monitor's LED indicator*, 2014 IEEE Joint Intelligence and Security Informatics Conference,IEEE, 2014, 264–267. Available from: https://ieeexplore.ieee.org/abstract/document/6975588.

12. A. Lopes and D. Aranha, *Platform-agnostic low-intrusion optical data exfiltration*, 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017), 2017, 474–480. Available from: http://dx.doi.org/10.5220/0006211504740480.

13. M. Guri, B. Zadov and Y. Elovici, *LED-it-GO: Leaking (a lot of) data from air-gapped computers via the (small) hard drive LED*, International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 2017, 161–184. Available from: http://arxiv.org/abs/1702.06715.

14.M. Guri, B. Zadov, A. Daidakulov, et al., xLED: Covert data exfiltration from air-gapped networks via router leds, *arXiv preprint arXiv*, (2017).

15.Z. Zheng, W. Zhang, Z. Yang et al., Exfiltration of data from air-gapped networks via unmodulated led status indicators, *arXiv preprint arXiv*, (2017).

16.M. Guri, D. Bykhovsky and Y. Elovici, Air-jumper: Covert air-gap exfiltration/infiltration via security cameras & infrared (IR), *Comput. Secur.,* **82** (2019), 15–29.

17.K. Jo, M. Gupta and S. K. Nayar, DisCo: Display-Camera Communication Using Rolling Shutter Sensors, *ACM Trans. Graphics.*, **35** (2016), 1–13.

18.H. Hao, L. Rujun, Q. Guolei et al., Covert-optical transmission channel based on LED display, *Commun. Technol.*, **51** (2018), 1689–1693.

19.M. Guri, O. Hasson, G. Kedma, et al., *An optical covert-channel to leak data through an air-gap* 2016 14th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2016. Available from: https://ieeexplore.ieee.org/document/7906933.

20.Kolb Helga, Much of the construction of an image takes place in the retina itself through the use of specialized neural circuits, in *How the Retina Works*, American Scientist, (2003), 28–35.

21.J. L. Ecker, G. S. Lall, S. Haq, et al., Melanopsin cells are the principal conduits for rod cone input to non-image-forming vision, *Nature*, **7191** (2008), 102–106.

22.G. Buchsbaum, An Analytical Derivation of Visual Nonlinearity *IEEE Trans. Biomed. Eng.,* **5**(1980), 237–242.

23.D. Mandal, K. Panetta and S. Agaian, *Human visual system inspired object detection and recognition*, 2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA), IEEE, 2012, 145–150. Available from:http://dx.doi.org/10.1109/TePRA.2012.6215669.

24.E. Simonson and J. Brozek, Flicker fusion frequency; background and applications, *Physiol. Rev.,* **32** (1952), 349–378.

25.S. D. Perli, N. Ahmed and D. Katabi, *PixNet: Interference-free wireless links using LCD-camera pairs,* 16th Annual Conference on Mobile Computing and Networking, MobiCom 2010 (2010), 1952, 137–148. Available from: http://dx.doi.org 10.1145/1859995.1860012.

26.T. Hao, R. Zhou and G. Xing, *COBRA: Color barcode streaming for smartphone systems*, Proceedings of the 10th international conference on Mobile systems, applications, and services, ACM, 2012, 85–98. Available from: http://dx.doi.org/10.1145/2307636.2307645.

27.W. Hu, *Lightsync: Unsynchronized visual communication over screen-camera links*, Proceedings of the 19th annual international conference on Mobile computing & networking, ACM, 2013, 15–26. Available from: http://dx.doi.org/10.1145/2500423.2500437.

28.T. Li, C. An, X. Xiao, et al., *Real-time screen-camera communication behind any scene* Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, ACM, 2015, 197–211. Available from: http://dx.doi.org/10.1145/2742647.2742667.

29.A. Wang, C. Peng, O. Zhang, et al., *InFrame: Multiflexing full-frame visible communication channel for humans and devices*, Proceedings of the 13th ACM Workshop on Hot Topics in Networks, ACM, 2014. Available from: http://dx.doi.org/10.1145/2670518.2673867.

30. A. Wang, Z. Li, C. Peng, et al., *Inframe++: Achieve simultaneous screen-human viewing and hidden screen-camera communication*, Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, ACM, 2015, 181-195. Available from: http://dx.doi.org/10.1145/2742647.2742652.

31. A. Costin, *Security of cctv and video surveillance systems: Threats, vulnerabilities, attacks, and mitigations,* Proceedings of the 6th international workshop on trustworthy embedded devices, ACM, 2016.Available from: https://dl.acm.org/citation.cfm?id=2995290.

AIMS Press