Authenticating Visual Cryptography Shares Using 2D Barcodes

Jonathan Weir and WeiQi Yan

Queen's University Belfast, Belfast, BT7 1NN, UK

Abstract. One of the problems pertinent with many visual cryptography (VC) schemes is that of authentication. VC provides a way of sharing secrets between a number of participants. The secrets are in the form of an image that is encoded into multiple pieces known as shares. When these shares are physically superimposed, the secret can be instantly observed. A known problem is that of authentication. How is it possible to know that the secret being recovered is genuine? There has been some work devoted to this using so called cheating prevention schemes which attempt to provide a means of traceability or authentication via a set of additional shares that are used to check authenticity. This paper proposes a scheme that attempts to alleviate this suspicion by using 2D barcodes as a means of authentication which may have more practicality in terms of real world usage. Results are provided using an application that is available on mobile devices for portable barcode reading.

1 Introduction

As far as secret sharing goes, visual cryptography [9] provides a very effective method for accomplishing this. One of the common problems that arise when designing VC schemes is whether or not the scheme can be cheated such that the set of known participants that are allowed to recover the secret can be cheated into recovering a secret of a different type without knowing they have been compromised.

Specific schemes that have been designed for cheat prevention focus on the probability that an attacker successfully cheats a scheme is negligible, that is, the known participants suspect that the shares or recovered secret is not genuine [4]. There are two types of approaches used when constructing these cheating prevention techniques. The first type is an authentication based method whereby each known participant is given an additional share that is used to authenticate the recovered secret. This provides the participants with the ability to verify the integrity of the shares before secret reconstruction takes place. The other authentication method uses a blind authentication technique that uses the properties of the reconstructed secret image. Blind authentication attempts to make it more difficult for the cheaters to predict the structure of a valid share that is in the possession of the qualified participants.

Despite the obvious advantages of having an additional share for verification purposes, the fact that an additional share is required is rather cumbersome and

Y.Q. Shi, H.J. Kim, and F. Perez-Gonzalez (Eds.): IWDW 2011, LNCS 7128, pp. 196–210, 2012.

impractical. This paper attempts to embed the verification information inside the recovered secret in the form of a 2D barcode. This way, no additional share is required and if cheating is suspected, the 2D barcode will not verify the invalid share after it is used to recover the secret. This is due to the fact that the barcode cannot be successfully guessed or that after a cheater has used his share, a barcode may not even be recovered as part of the secret.

Additionally, an extended form of this secret sharing is presented which embeds the 2D barcode as part of the cover image for each participant. This allows share verification before the secret recovery has even been attempted. This type of early verification can take place using mobile devices such as iPhones, which support software barcode readers that use the mobile devices onboard camera.

The main contributions discussed within this paper are:

- Using a 2D barcode to authenticate a VC share.
- Any VC scheme can benefit from this type of authentication, depending on the practical requirements.
- The 2D barcode can be used as the secret transport mechanism. That is, a long string of alphanumeric characters can be embedded inside the barcode.
- This increases the overall capacity of sharing a large amount of data within a small manageable set of shares.
- The practical usage involving mobile devices with an onboard camera is very simple and effective.

The remainder of the paper is organized accordingly, Section 2 provides a brief background on cheating and cheat prevention schemes in VC and the related work accomplished in this area, Section 3 highlights the proposed idea and how it will be achieved, while Section 5 provides the conclusion.

2 Related Work

Despite visual cryptography's secure nature, many researchers have experimented with the idea of cheating the system. Methods for cheating the basic VC schemes have been presented, along with techniques used for cheating extended VC schemes [7,10,18].

Prevention of cheating via authentication methods [10] has been proposed which focus on identification between two participants to help prevent any type of cheating taking place. Yang and Laih [18] presented two types of cheating prevention, one type used an online trust authority to perform the verification between the participants. The second type involved changing the VC scheme whereby the stacking of two shares reveals a verification image, however this method requires the addition of extra pixels in the secret.

Another cheating prevention scheme described by Horng et al. [7], through which if an attacker knows the exact distribution of black and white pixels of each of the shares of honest participants then they will be able to successfully attack and cheat the scheme. Horng's method prevents the attacker from obtaining this distribution. Since then, there have been numerous efforts devoted to designing cheating prevention schemes within visual cryptography [5,8,12].

Successfully cheating a visual cryptography scheme (VCS) however, does not require knowledge of the distribution of black and white pixels. Hu and Tzeng [8] where able to present numerous cheating methods, each of which where capable of cheating Horng et al.'s cheating prevention scheme. Hu and Tzeng also present improvements on Yang and Laih's scheme and finally present their own cheating prevention scheme which attempts to minimize the overall additional pixels which may be required. No online trust authority is required and the verification of each image is different and confidential. The contrast is minimally changed and the cheating prevention scheme should apply to any VCS. Hu and Tzeng where also able to prove that both a malicious participant (**MP**), that is $\mathbf{MP} \in P$, and a malicious outsider (**MO**), $\mathbf{MO} \notin P$, can cheat in some circumstances, where P is the set of participants.

The **MP** is able to construct a fake set of shares using his genuine share. After the fake share has been stacked on the genuine share, the fake secret can be viewed. The second cheating method involving an **MO** is capable of cheating the VC scheme without having any knowledge of any genuine shares. The **MO** firstly creates a set of fake shares based on the optimal $(2, 2)$-VCS. Next, the fake shares are required to be resized to that of the original genuine shares size. However, an assumption is to be made on the genuine shares size, namely that these shares where printed onto a standard size of paper, something like A4 or A3. Therefore, shares of those sizes are created, along with fractions of those sizes. Management of this type of scheme would prove to be problematic due to the number of potential shares created in order to have a set of the correct size required to cheat a specific scheme, but once that size is known, cheating is definitely possible as an **MO**.

A traceable model of visual cryptography [1] was also examined which endeavours to deal with cheating. It deals with the scenario when a coalition of less than k traitors who stack their shares and publish the result so that other coalitions of the participants can illegally reveal the secret. In the traceable model, it is possible to trace the saboteurs with the aid of special markings. The constructions of traceable schemes for both (k, n) and (n, n) problems were also presented. Furthermore, other practical applications have also been examined including one involving biometrics [3,6,14,15,16,17]. This paper further adds to this list of practical applications involving visual cryptography.

3 The Proposed Scheme

With the previous work in mind, a number of areas should be focused on during the creation of the new scheme. A suitable barcode must be selected to perform the authentication along with a VC scheme capable of processing and handling this type of information. Using this type of technique with existing VC schemes should also be considered. This would allow an authentication mechanism to be used with current techniques. The actual problem of authentication and the issues facing it should be acknowledged as well.

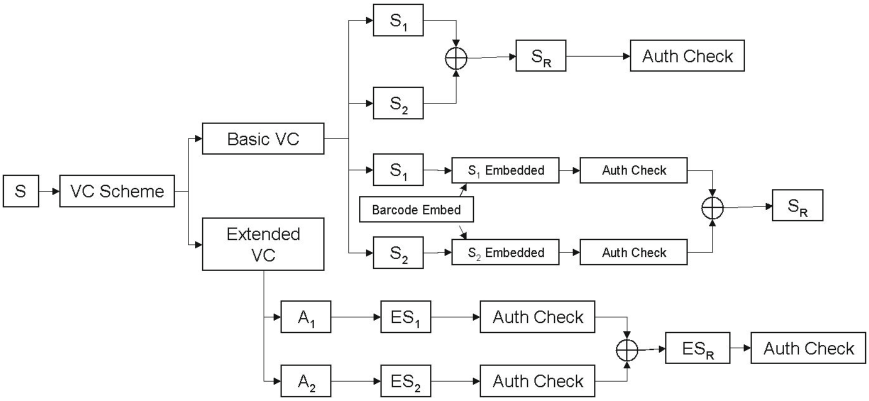


Fig. 1. Flowchart of the proposed VC authentication system

Figure 1 provides the flowchart for the proposed scheme. Two separate VC schemes are used, a traditional basic VC scheme and an extended VC scheme.

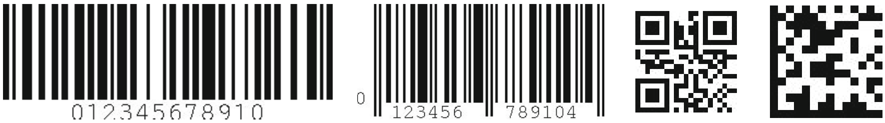
The secret S is selected. For the traditional scheme, this would be the barcode that is used for authentication. The extended scheme can use this type of secret as well, or another secret such as a PIN number or code number for a safe or bank vault. After the secret has been input, the choice of VC scheme is next. The basic VC scheme will go on to process the secret and generate two shares S_1 and S_2. Physically superimposing these shares will recover the secret image S_R. This results in the recovery of the barcode which can then be used for authentication. The basic scheme also offers the ability to authenticate each share for traditional secret recovery. The barcodes disappear after the secret has been recovered.

The extended scheme functions quite differently, in that more authentication checks are possible during the process due to the nature of the extended VC scheme. Cover images A_1 and A_2 are generated. These are barcodes which contain unique authentication information. Using these authentication images, two shares ES_1 and ES_2 are created which when combined can recover the original secret ES_R. It can be noticed that further authentication checks are now possible before secret recovery takes place. The final secret that is recovered can also be checked for authenticity.

3.1 Determining a Suitable Barcode

A barcode is an optical machine-readable representation of data, which shows data about the object to which it attaches. Traditionally, barcodes represent data by using parallel lines with varying widths and spacing. Other geometric patterns, such as rectangles, dots and hexagons have also be used in the creation of barcodes. Barcode scanners are used to read the barcode and decode the information within it.

There are a vast array of barcode types to choose from, but the main examples used within this paper include the more common types: Code-128 (1D), EAN-13 (1D), QR code (2D), Datamatrix (2D). An example of these types of barcode are shown in Figure 2.



(a) Code-128 (b) EAN-13 (c) QR code (d) Datamatrix

Fig. 2. Example barcode types. Each of which contain the same information string: 012345678910

However, for this type of application, alphanumeric characters are more suited to the type of secret verification that will be used, so EAN-13 will not be considered during the testing. It is included here merely as an example.

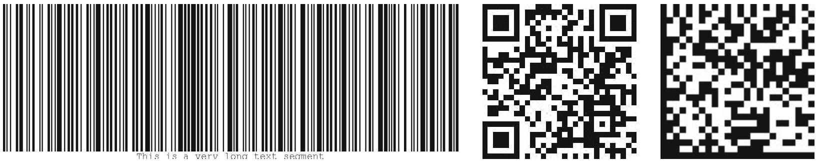
Barcodes are highly robust when it comes to extracting the information contained within them. Using mobile devices equipped with a camear, barcodes, in the form of QR codes are a common way to read information from a magazine or advertisement [11]. Such mobile devices have limited processing power and low resolution cameras. This requires the barcodes to be machine readable on limited devices. This is why they are useful for the work combining visual cryptography.

Traditionally, visual cryptography deals with binary images, this is another great advantage for combining it with barcodes. No colour image processing is required and the share generation can use many types of existing VC techniques to generate the required shares. Ranging from schemes that expand the pixels into a 2 × 2 block to size invariant schemes which maintain the original aspect ratio of the secret. Barcode readers are still capable of recovering the information from the barcode, even after such a distortion may have taken place.

Another issue which crops up time and again is that of capacity. Clearly hiding and recovering quantities of text within an image, especially one that has been encoded using visual cryptography has been problematic. Using such methods can be cumbersome and difficult to read. Recovering barcodes accurately which store long types of textual information would be a much more useful and easier. Figure 3 illustrates a number of barcode schemes which contain a long string of text. Each barcode can be accurately read by a barcode reader.

Share size is another issue that researchers face when designing visual cryptography schemes. Management of large shares is unwieldy and the smaller the share, the better it is for the application. This goes hand in hand with secret recovery. If the shares are small, the secret will be unclear, unless the secret is in the form of a barcode, such as a QR code, which can contain a large amount of textual data inside a small image.

From the barcodes displayed in Figure 2 and 3 it can be observed that the Code-128 barcode is the most effected by the change in information. The final image size jumped from 303 × 106 to 1161 × 392. The QR code and Datamatrix



(a) Code-128 (b) QR code (c) Datamatrix

Fig. 3. Textual data within each of the barcodes that support that type of information. The information string: "This is a very long text segment".

Table 1. Table of resolutions depending on the barcode type used along with the amount and type of data represented

Barcode type	Resolution (digits only)	Resolution (text, including spaces)
Code-128	303 × 106	1161 × 392
QR code	63 × 63	87 × 87
Datamatrix	70 × 70	120 × 120

both fared much better, with less of a size increase. Both of which remained at a very manageable size of 87 × 87 and 120 × 120 respectively. These resolution increases are tabulated in Table 1.

Based on this, QR code and Datamatrix representation will be used during each of the test phases when combining these barcode types with visual cryptography as an authentication medium. Improving efficiency in terms of processing power, when generating shares, favours smaller secret images to begin with. So we can successfully ignore the Code-128 type of barcode from the remainder of the testing.

3.2 Visual Cryptography Scheme Selection

Now that the type of barcode has been chosen, the next step involves using a VC scheme that allows sufficient secret recovery such that a barcode scanner can be used to read the recovered barcode. This is extremely important, if the barcode cannot be read correctly, the shares or the participants cannot be authenticated. Due to the small nature of the QR code image that is produced, many VC schemes would be well suited to sharing this type of information. A number of tests will be performed using a variety of schemes to illustrate this.

The QR code image that will be used as an authentication image can be viewed in Figure 4. The authentication message that goes along with it is also included. The process in itself will be one of a digital nature at first, using a computer to recover the secret and then test the barcode for authentication. The tests will then be extended into the physical form of testing, so that tests can be done using traditional means with a camera on a mobile device.

The application used to read the barcode is an open source software suite known as ZBar [2]. This package is used because it supports a multitude of operating systems which includes the iPhone and other embedded devices. Mobile

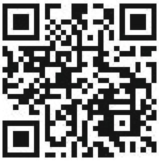
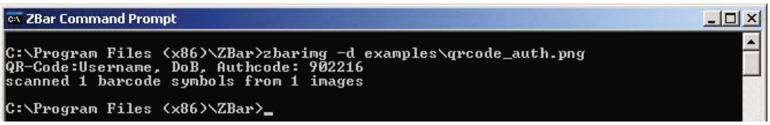


Fig. 4. QR code (87 × 87) with personal details and the authentication number. Embedded text: "Username, DoB, Authcode: 902216"



```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_auth.png
QR-Code:Username, DoB, Authcode: 902216
scanned 1 barcode symbols from 1 images

C:\Program Files (x86)\ZBar>_
```

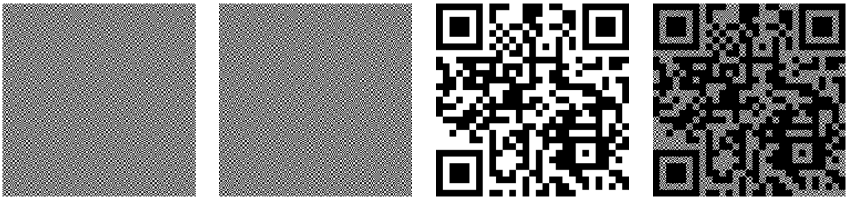
Fig. 5. ZBar reading the QR code and confirming the correct details

devices such as iPhones are becoming evermore popular, this is a good indication as to whether this research has merit in a real world application. Figure 5 provides an example of ZBar reading the corresponding QR code in Figure 4 and returning the correct details.

Essentially there are two different types of secret recovery, the first method will use the XOR binary operation to combine the shares, the later uses the OR operation [13]. XOR can be used when dealing primarily in a digital environment, as it removes any grey shading that may be observed when black and white pixels are arranged together one after another, leaving either solid white or solid black sections of the image.

A number of VC schemes will be tested throughout. A basic $(2,2)$ VC scheme which involved 2×2 pixel expansion and a size invariant scheme, both $(2,2)$ and $(2,3)$ to show that k out of n sharing is possible. Figure 6 provides an example of the shares based on (2,2) VC scheme, with a pixel expansion of 2×2. Figure 7 gives an example based on a (2,2) size invariant scheme. Included within the figures are the XOR and OR secret recovery images.

Figure 8 provides the results of a (2,3) size invariant scheme which shows that the same results are possible using a k out of n secret sharing scheme.



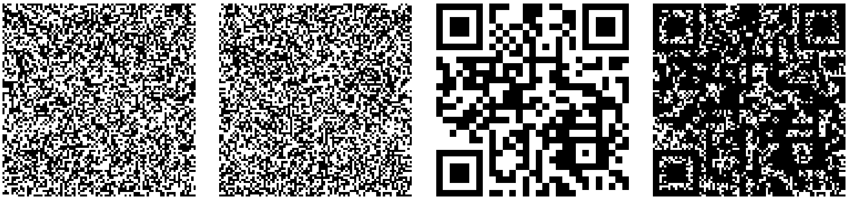
(a) QR code share one (174 × 174)

(b) QR code share two (174 × 174)

(c) XOR secret recovery (174 × 174)

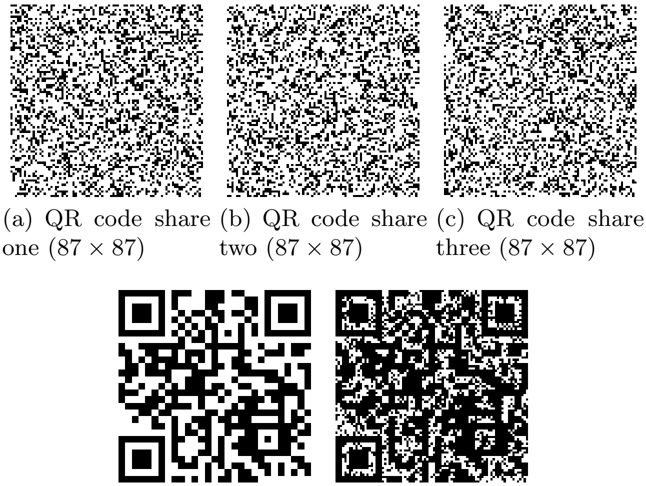
(d) OR secret recovery (174 × 174)

Fig. 6. A $(2,2)$ basic visual cryptography secret recovery process involving the XOR and OR binary operations



(a) QR code share one (87 × 87) (b) QR code share two (87 × 87) (c) XOR secret recovery (87 × 87) (d) OR secret recovery (87 × 87)

Fig. 7. A $(2, 2)$ size invariant visual cryptography secret recovery process involving the XOR and OR binary operations



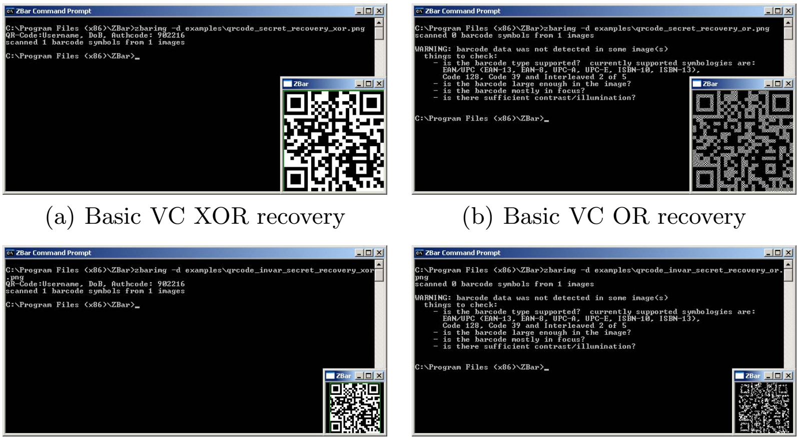
(a) QR code share one (87 × 87) (b) QR code share two (87 × 87) (c) QR code share three (87 × 87)

(d) XOR secret recovery (87 × 87) (e) OR secret recovery (87 × 87)

Fig. 8. A $(2, 3)$ size invariant visual cryptography secret recovery process involving the XOR and OR binary operations

Both the recovered secrets using the XOR methods can be verified very easily by the barcode reader application. However, the barcode recovered using the OR operation presents a problem, both barcodes cannot be successfully read. There are a number of reasons why this is the case. The barcode could possibly be too small. This can be ruled out, as the barcode is twice as large, or the same size as the original barcode. The focus on the image is another issue. This is not the case either, as the testing so far has been digital only. The last reason why the barcode may not be read is that there may not be sufficient contrast or illumination on the image.

Figure 9 shows the results of attempting to recover the authentication data from each of the secrets. It can be observed that the barcodes recovered using the XOR operation can be perfectly read, whereas the secrets recovered using the OR operation cannot be detected. This helps to reinforce the point about



```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_secret_recovery_xor.png
QR-Code:Username, DoB, Authcode: 902216
scanned 1 barcode symbols from 1 images

C:\Program Files (x86)\ZBar>_
```

(a) Basic VC XOR recovery

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_secret_recovery_or.png
scanned 0 barcode symbols from 1 images

WARNING: barcode data was not detected in some image(s)
  things to check:
    - is the barcode type supported?  currently supported symbologies are:
      EAN/UPC (EAN-13, EAN-8, UPC-A, UPC-E, ISBN-10, ISBN-13),
      Code 128, Code 39 and Interleaved 2 of 5
    - is the barcode large enough in the image?
    - is the barcode mostly in focus?
    - is there sufficient contrast/illumination?


C:\Program Files (x86)\ZBar>_
```

(b) Basic VC OR recovery

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_invar_secret_recovery_xor
.png
QR-Code:Username, DoB, Authcode: 902216
scanned 1 barcode symbols from 1 images

C:\Program Files (x86)\ZBar>_
```

(c) Size invariant VC XOR recovery

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\qrcode_invar_secret_recovery_or.
png
scanned 0 barcode symbols from 1 images

WARNING: barcode data was not detected in some image(s)
  things to check:
    - is the barcode type supported?  currently supported symbologies are:
      EAN/UPC (EAN-13, EAN-8, UPC-A, UPC-E, ISBN-10, ISBN-13),
      Code 128, Code 39 and Interleaved 2 of 5
    - is the barcode large enough in the image?
    - is the barcode mostly in focus?
    - is there sufficient contrast/illumination?


C:\Program Files (x86)\ZBar>_
```

(d) Size invariant VC OR recovery

Fig. 9. Attempted authentication of each of the barcodes based on the type of secret recovery used

share tampering. If the shares have been altered, then reading of the authentication information from the barcode becomes problematic. This will give a good indication as to whether the shares have been altered or not.

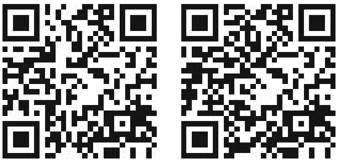
This is an important point as many VC schemes rely on a difference in contrast in order to display the secret after recovery. This is why the XOR operation is much better suited to this type application. From Figure 9 it can be seen due to the dark nature of the secrets recovered using the OR operation, authentication becomes difficult. The barcode reader has difficulty in determining where the barcode is on the image. Additionally, size is not an issue, as the barcode can be successfully read after the XOR recovery on the original barcode secret that was encoded using a size invariant VC scheme.

3.3 The Authentication Problem

After observing the results obtained with the previous example, the authentication problem can be viewed from a simpler stance. Slight alterations create or generate barcodes that are completely unreadable or could possibly generate valid barcodes with invalid data.

Figure 10 illustrates this with a simple example, which increments the last digit of the authentication string by one. Viewed side by side, it can be seen that a superficial change in the data generates a vastly different barcode. This is important in terms of authentication. If the shares are tampered with and a barcode is generated that is not exact, no authentication will be possible. This renders the shares useless. This is a great property to have for authentication purposes.

This favours VC a lot from the point of view that if additional noise is added to the tampered image then a false or non-genuine authcode will be extracted,

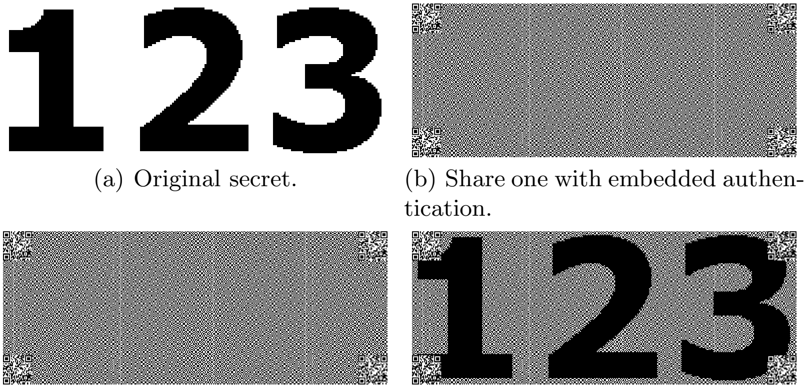


(a) Authcode: 1111 (b) Authcode: 1112

Fig. 10. Two QR codes, both containing a similar authcode. A slight difference in authcode produces a vastly different QR code.

highlighting the fact that the shares cannot be trusted for that participant. Minor disruptions to the shares which result in spurious or completely invalid barcodes being generated is a good way to keep track of authenticate shares.

Figure 11 shows how the shares can be authenticated using the 2D barcode embedded into the corner of the share. The original image included along with its corresponding shares and recovered secret. The barcode share is embedded with a similar density and distribution as the share it is placed into. It also disappears when the secret is recovered. Allowing for more exact recovery.



(a) Original secret.

(b) Share one with embedded authentication.

(c) Share two with embedded authentication.

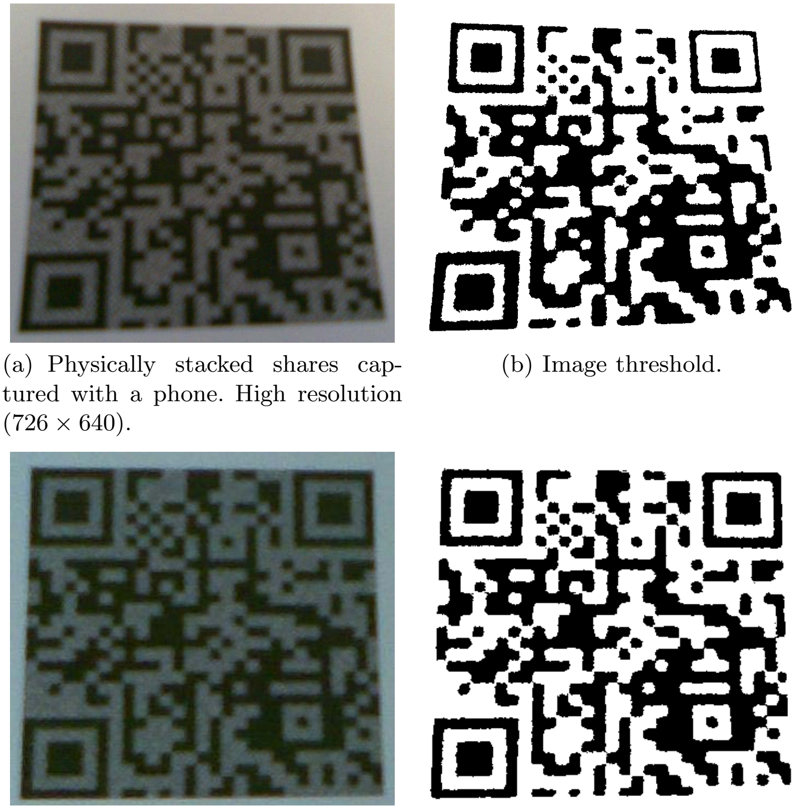
(d) Recovered secret.

Fig. 11. Using a 2D barcode to authenticate each share

3.4 Authenticating the Shares

Authenticating the shares at a practical level using the techniques described can be achieved using a standard mobile device or smart phone. Many of these devices support programming platforms such as Python. This is very useful when it comes to simple image processing that may be required to process photographs of the shares when it comes to authenticating them.

Figure 12 provides an example of a photograph of the same barcode taken at different resolutions (Figure 12(a) and 12(c)). The angles at which the photographs were taken also differs in each of the figures. This helps to reinforce the robustness of using a barcode for this type of authentication. Barcodes are very resilient to changes in angle such as in this figure. It also removes the onus from the user of having to take a photograph at a specific angle or to carefully align the camera. The thresholds of each of the barcode photographs have been taken, these are visible in Figure 12(b) and 12(d) respectively.



(a) Physically stacked shares captured with a phone. High resolution (726 × 640).

(b) Image threshold.

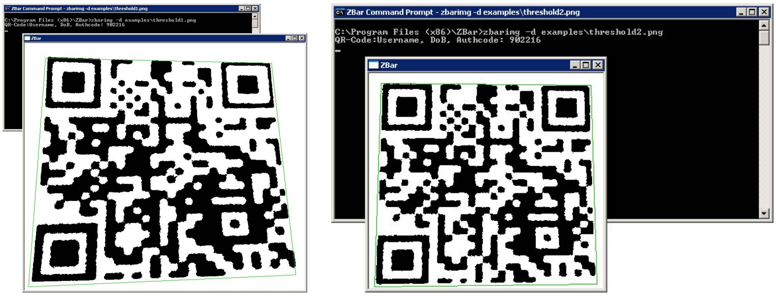
(c) Physically stacked shares captured with a phone. Low resolution (354 × 325).

(d) Image threshold.

Fig. 12. Capturing the physically stacked barcode images and thresholding them so that the binary barcode reader can process them correctly for authorization

Due to the noisy and grainy nature of the photographs, taking a threshold of the original photo is necessary when it comes to accurately reading the barcode. Despite the fact that barcodes are very robust to many types of image manipulation, contrast and brightness are also important factors in recovery and reading of a barcode. If contrast and brightness conditions are suboptimal, the code will not be successfully read.

The Zbar application can then be used to read the barcode from each of the threshold images that are processed. Figure 13 illustrates this recovery process, displaying clearly each of the authentication messages. The application also correctly outlines the area that it is reading and has recognized as the barcode itself. The resolution and angle of the photograph are not overly important to the barcode reader as is illustrated by the successful recovery of the barcode after the shares have been physically stacked.


ZBar Command Prompt - zbarimg -d examples\threshold1.png
ZBar
ZBar Command Prompt - zbarimg -d examples\threshold2.png
ZBar

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\threshold1.png
QR-Code:Username, DoB, Authcode: 902216
```

```
C:\Program Files (x86)\ZBar>zbarimg -d examples\threshold2.png
QR-Code:Username, DoB, Authcode: 902216
```

(a) Successful barcode recovery 1.

(b) Successful barcode recovery 2.

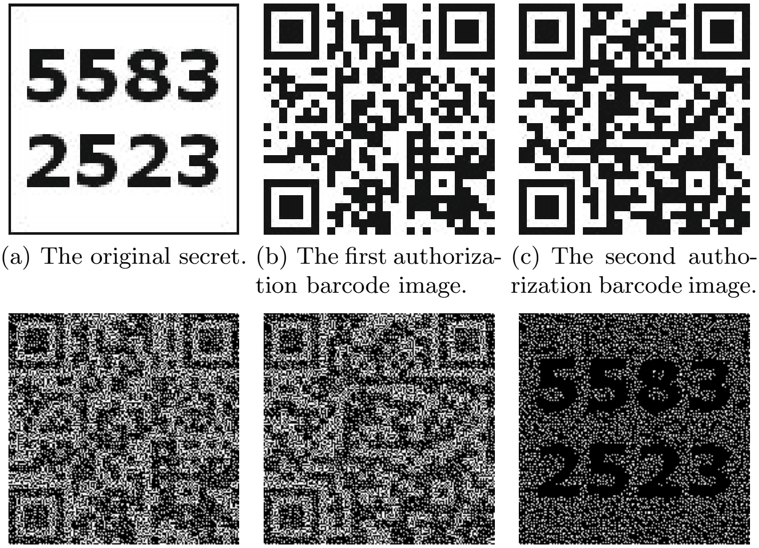
Fig. 13. Recovering the barcode information from each of the images processed by the phone

Furthermore, the authcode that has been extracted from each of the share can then be checked against a database on the mobile device or against a remote database on the network or internet for an additional check for the correct details.

Including a verification process for each individual share is also achievable. Using an extended form of visual cryptography, the authorization barcodes can be used as the secrets for each share. The shares can be verified and checked using a mobile device in the same manner as the previous example. Each share can have the same authcode or a unique code, depending on the type of authentication required by each person. Another advantage of this is that after verification, the main secret (a safe combination for example) can be completely recovered, while having no part of the original authorization barcode obscuring it.

Figure 14 provides an example of how the extended type of visual cryptography shares can be used to achieve this. The secret can be viewed in Figure 14(a). The authorization images are shown within the Figure 14(b) and 14(c). The shares used for the secret recovery can be observed in Figure 14(d) and 14(e). These shares contain the verification barcodes from Figure 14(b) and 14(c) respectively. The recovered secret is displayed in Figure 14(f). The recovery occurs when each of the shares physically superimposed.

One issue with this type of authorization is that changes in light intensity and contrast have a big impact on reading the barcode embedded within the share.



(a) The original secret. (b) The first authorization barcode image. (c) The second authorization barcode image.

(d) Extended share one. (e) Extended share two. (f) Secret recovery of the original secret.

Fig. 14. Extended VC with an authentication mechanism built into the share images

This is an area where improvement can be difficult. If too much of the barcode is visible, reconstruction of the secret could be obscured by this.

4 Security Analysis

The security of the scheme rests entirely with VC and its construction. Firstly, the presented scheme is secure in that given any amount of sub-pixels from a single share, it is impossible to tell if the corresponding shares sub-pixels represent a black or a white pixel after superimposing them. This can be illustrated using a probabilistic proof. For a random secret, it cannot be assumed that the pixel values selected to represent those from the secret are uniformly distributed. This is down to the size invariant scheme, in that one pixel from the secret has to be represented by one pixel from one of the shares, while the second share must also contain just a single pixel while keeping the value of the secret pixel hidden.

If a black pixel is to be represented from the secret, then its corresponding pattern is always black. Conversely, if a white pixel is to be represented then it can have two possible representations. A white pixel in each share is possible, or a white pixel in share one with a black pixel in the second share, which ultimately ends up as a black pixel, but does indeed represent a white pixel from the secret. So if a pixel is examined and found to be black in one share, the probability that it is black in the second share is 0.5. However, if the pixel in the share is white then there is also a probability of 0.5 that the pixel will be either black or white.

This makes it very difficult to analyze the secret based on these probabilities due to the nature of the pixel representations.

5 Conclusion

The principal idea from this paper is to use barcodes as an authentication means by which visual cryptography shares can be verified. The type of applications that can make use of this secret sharing are many. Using the scheme as a verification for opening bank vaults or other security related schemes is an important issue and can be achieved with relative simplicity in terms of checking that the barcode is accurate and untampered. Simple, manageable methods of share verification are quite difficult and require other methods as previous described, such as other shares.

This paper differs in that it presents the verification data within the share in the form of a barcode which helps to remove the issues that plague other schemes. Especially since such common devices such as mobile telephones and smart phones (iPhone) can be used as a means to facilitate this, the additional hardware requirement is not something of an issue.

From the results it can be observed that while the barcode application may not be able to read the shares that have been digitally superimposed, the shares that are physically stacked and captured with a camera can be recovered and identified successfully. This would be a good way of verifying and authenticating a set of shares to confirm the identity of a particular party or individual.

Along with that, storing a larger amount of textual data inside the barcode and read using a mobile device has also been accomplished. Storing a long, easily readable and easily recoverable secret using traditional VC techniques becomes very problematic, as the share size increases dramatically as more text is added. Using the scheme presented within this paper, the problem of share size is removed completely in terms of the amount of data can be held inside the embedded barcode. Essentially, smaller shares with more information are a great improvement.

References

1. Biehl, I., Wetzel, S.: Traceable Visual Cryptography. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 61–71. Springer, Heidelberg (1997)
2. Brown, J.: ZBar bar code reader, `http://zbar.sourceforge.net/`
3. Chan, C.-W., Lin, C.-H.: A New Credit Card Payment Scheme Using Mobile Phones Based on Visual Cryptography. In: Yang, C.C., Chen, H., Chau, M., Chang, K., Lang, S.-D., Chen, P.S., Hsieh, R., Zeng, D., Wang, F.-Y., Carley, K.M., Mao, W., Zhan, J. (eds.) ISI Workshops 2008. LNCS, vol. 5075, pp. 467–476. Springer, Heidelberg (2008)
4. Chang, C.C., Chen, T.H., Liu, L.J.: Preventing cheating in computational visual cryptography. Fundamenta Informaticae 92, 27–42 (2009)
5. De Prisco, R., De Santis, A.: Cheating immune threshold visual secret sharing. The Computer Journal 53, 1485–1496 (2010), `http://dx.doi.org/10.1093/comjnl/bxp068`

6. Hegde, C., Manu, S., Shenoy, P., Venugopal, K., Patnaik, L.: Secure authentication using image processing and visual cryptography for banking applications. In: 16th International Conference on Advanced Computing and Communications, pp. 65–72 (December 2008)
7. Horng, G., Chen, T., Tsai, D.S.: Cheating in visual cryptography. Designs, Codes and Cryptography 38(2), 219–236 (2006)
8. Hu, C.M., Tzeng, W.G.: Cheating prevention in visual cryptography. IEEE Transactions on Image Processing 16(1), 36–45 (2007)
9. Naor, M., Shamir, A.: Visual Cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
10. Naor, M., Pinkas, B.: Visual Authentication and Identification. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 322–336. Springer, Heidelberg (1997)
11. Rouillard, J.: Contextual qr codes. In: ICCGI 2008. The Third International Multi-Conference on Computing in the Global Information Technology, July 27-August 1, pp. 50–55 (2008)
12. Tsai, D.S., Chen, T.H., Horng, G.: A cheating prevention scheme for binary visual cryptography with homogeneous secret images. Pattern Recognition 40, 2356–2366 (2007), http://portal.acm.org/citation.cfm?id=1240339.1240571
13. Tuyls, P., Hollmann, H.D.L., Lint, J.H.V., Tolhuizen, L.: XOR-based visual cryptography schemes. Designs, Codes and Cryptography 37, 169–186 (2005), 10.1007/s10623-004-3816-4
14. Tuyls, P., Kevenaar, T., Schrijen, G.-J., Staring, T., van Dijk, M.: Visual Crypto Displays Enabling Secure Communications. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 271–284. Springer, Heidelberg (2004)
15. Weir, J., Yan, W.: Resolution variant visual cryptography for street view of google maps. In: IEEE International Symposium on Circuits and Systems, ISCAS 2010 (May 2010)
16. Weir, J., Yan, W.-Q.: Dot-Size Variant Visual Cryptography. In: Ho, A.T.S., Shi, Y.Q., Kim, H.J., Barni, M. (eds.) IWDW 2009. LNCS, vol. 5703, pp. 136–148. Springer, Heidelberg (2009)
17. Weir, J., Yan, W.: Image hatching for visual cryptography. In: Proceedings of the International Machine Vision and Image Processing Conference, pp. 59–64. IEEE Computer Society Press, Los Alamitos (2009)
18. Yang, C., Laih, C.: Some new types of visual secret sharing schemes. In: National Computer Symposium (NCS 1999), vol. III, pp. 260–268 (December 1999)