

Secure E-Payment Method Based on Visual Cryptography

Trihastuti Yuniati

School of Electrical Engineering and Informatics
Institute of Technology Bandung
Bandung, Indonesia
trihastuti.yuniati@gmail.com

Rinaldi Munir

School of Electrical Engineering and Informatics
Institute of Technology Bandung
Bandung, Indonesia
rinaldi@informatika.org

Abstract—In recent time there is a rapid growth in the online shopping market. With ever increasing popularity of the online shopping, debit or credit card fraud and personal information security are major concerns. Phishing and identity theft are common threats to online shopping. Phishing is a method of stealing personal confidential information, such as username, password, credit card details, from legitimate user. In this paper, we have proposed a credit card payment method using visual cryptography. Visual cryptography is applied to captcha generated by the merchant during the registration phase, to a text file containing credit card details sent by the Merchant Plug-In to the Card Provider Directory Server, and to Quick Response Code containing One Time Password that is used to authorize payment transaction. The purpose of this method is to improve e-payment security, especially in terms of authentication, authorization, and confidentiality. This method provides limited credit card details to the merchant and prevents phishing and identity theft so that the customer's confidence will increase.

Keywords—captcha, e-payment, one time password, online shopping, quick response code, visual cryptography

I. INTRODUCTION

With increasingly widespread use of the internet in everyday life has changed the human habits and preferences. In the past, people made direct face-to-face transactions between sellers and buyers. Nowadays, they prefer to shop over the internet or known as online shopping. As a result, electronic payment systems are becoming more essential in modern businesses. However, there have been a large number credit card fraud incidents over the internet due to the security weakness of the electronic payment system.

Phishing and identity theft are the online shopping's common threats. Phishing is a method of stealing personal confidential information from victims. Victims are tricked into providing their credential by a combination of spoofing techniques. A number of solutions have been proposed in past to prevent this problem, but they are still not effective enough to stop the problem from happening. In this paper, a new approach to securing online payment system using visual cryptography has been proposed. Visual cryptography will be applied to three processes: user registration to the merchant's site, credit card verification through the card provider, and during payment authorization process between the user, merchant, and card issuer.

II. VISUAL CRYPTOGRAPHY

Visual cryptography is a secret sharing scheme where it is an encryption technique to hide information in an image in such a way that it can be decrypted by superimposing two or more shares. Share is a random pixel image generated using visual cryptography algorithm. This scheme was developed by

Naor and Shamir in 1994. The basic model consists of a printed page of cipher text and a printed transparency which serves as a secret key. The original clear text is revealed by placing the transparency with the key over the page with cipher text, even though each of them is indistinguishable from random noise [3].

In scheme (2, 2), visual cryptography produces two shares of the same image, one image contains random pixels and the other contains secret information. In this scheme, we consider black and white image having a binary resolution i.e. white pixel means 0 and black pixel means 1. We consider 2x2 matrix for each pixel in a given image. A single pixel will have 2 matrices. One matrix will be randomly selected and the other will be generated according to the pixel color i.e. black or white pixel. In scheme (2, 2) with 2-subpixels expansion, each single pixel of the original image is expanded to 2-subpixels in shared image. Figure 1 shows illustration of scheme (2, 2) with 2-subpixels expansion [4].












Pixel	Prob.	Shares #1 #2	Superposition of two shares
	$p = 0.5$	 	 White pixel
	$p = 0.5$	 	
	$p = 0.5$	 	 Black pixel
	$p = 0.5$	 	

Fig. 1. (2, 2) visual cryptography scheme with 2-subpixels expansion

No share leads to the original image pixel because every time random pixel are encrypted to create a secret image. When the two shares are superimposed on each other, the value of the original pixel can be determined.

There are many variations of visual cryptography. Based on the visual secret sharing scheme, there are k out of k (k, k) and k out of n (k, n) solutions. Based on the processed image, there are black and white images or color images. Based on the secret file types, there are text-based, image-based, and extended visual cryptography implemented in the QR Code.

III. RELATED WORKS

A brief survey of related work in the field of anti-phishing and secure online payment frameworks based on visual cryptography are presented in this section. A visual cryptography based anti-phishing mechanism was proposed in [1]. It is uses a graphical captcha that generates two shares. The shares are used as part of authentication. Only genuine users can provide this shares. An online payment system using steganography and visual cryptography is presented in [7], but

the paper doesn't focus on phishing. There is no way to detect whether the site is a spoofed website or a legitimate website. Participants involved in the proposed system are user, merchant, bank, and Certified Authority (CA), but CA has difficulties in passing information to which bank, and there is no validation between user and CA. Similar approaches are presented in [4], [5], and [6], but in this system user and merchant must be registered on the bank server, even though in the existing system merchant do not always have direct cooperation with the card issuer, but the card provider.

IV. PROPOSED FRAMEWORK

For detecting and preventing phishing and identity theft in order to obtain a more secure credit card payment system in terms of authentication, authorization, and confidentiality, a new approach is proposed in this paper. This secure e-payment method is based on visual cryptography. The participants involved are cardholder (user), merchant, acquirer, card issuer, and card provider. There are three main phases in this proposed method, registration phase, login and purchasing phase, and payment phase. Figure 2 shows registration phase flowchart.

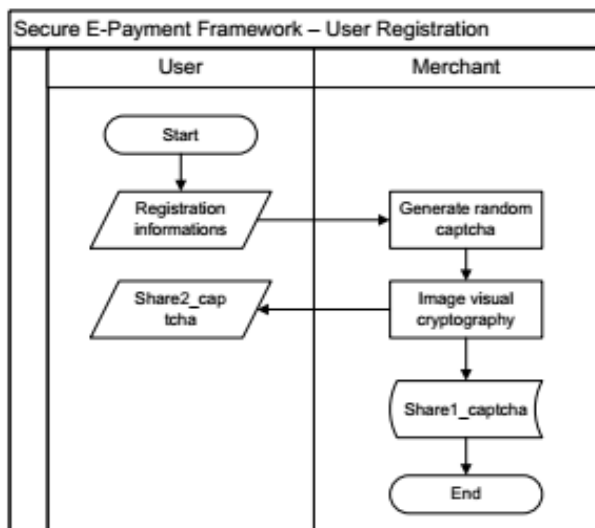


Fig. 2. Registration phase flowchart

As can be seen in Figure 2, operations in the registration phase are presented. In this phase, merchant generated a random captcha image that appears in the registration form. On that page, user enters his/her registration data, including username, password, and captcha string. If all of the user's entries are valid, then the merchant generated two shares from captcha image. One share is saved in the merchant's database, and the other is sent to the user via email.

In the login phase, the actual authentication occurs. This process is built in such a way that it can detect phishing attack. In the login page, user gives his/her credentials. The merchant's server compares username and password with the data stored in the database. If they are appropriate, the user will be directed to share validation page. User then gives his/her captcha share obtained from the registration phase. The merchant's server combines the user's share with the share stored in database. By stacking each other, if the user's share is valid, then the original captcha is established. Figure 3 shows login and purchasing phase flowchart.

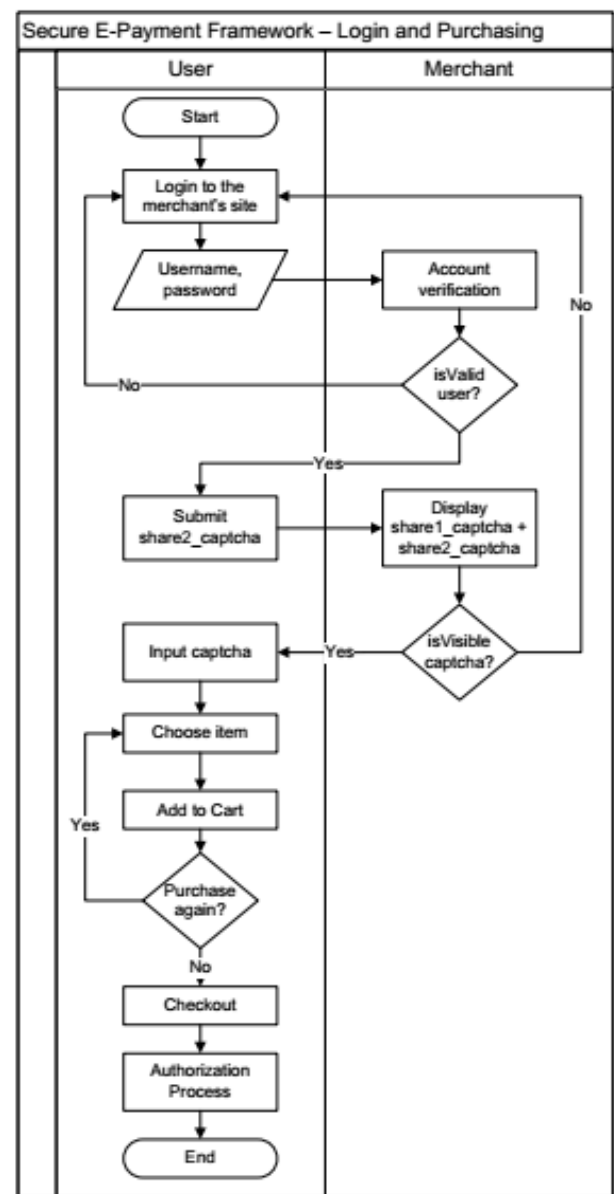


Fig. 3. Login and purchasing phase flowchart

The original captcha can find whether the user is really genuine user or a phishing attack is carried out. If the user's share is invalid, the captcha image will not be visible and the user will not be able to login to the merchant's site. In the other hand, if the merchant's captcha is invalid (that means phishing attack has occurred), the captcha string will also not be visible so the user will be able to know that the site is fake. This login mechanism can identify phishing activity and prevent it efficient with 100% true positives.

If both user and merchant is valid, then the captcha string will be visible in the share validation page. The user is then asked to enter the captcha string that is visible. After login to the merchant's site, then the user may purchase goods. User chooses items he/she wants to buy, adds them to the cart, inputs order details, and makes payment.

The payment phase is divided into 2 sub-phases, checkout phase and authorization process phase. Figure 4 and Figure 5 shows checkout phase flowchart. In the checkout process in Figure 4, user interacts with Merchant Plug-In (MPI), Card Provider Directory Server, (CPDS), and Access Control

Server (ACS). MPI is a software module integrated with merchant's website, used to provide the interface between the card-provider's program and the merchant's payment processing software. CPDS is a server operated by the card provider to route authentication requests from the merchants to the card issuers' ACSs and to return the results of authentication. ACS is a server with registered user account and access information. It is operated by the card issuer or its processor. It validates user participation in the program, performs user verification at time of purchase, and provides digitally signed responses to merchants [8].

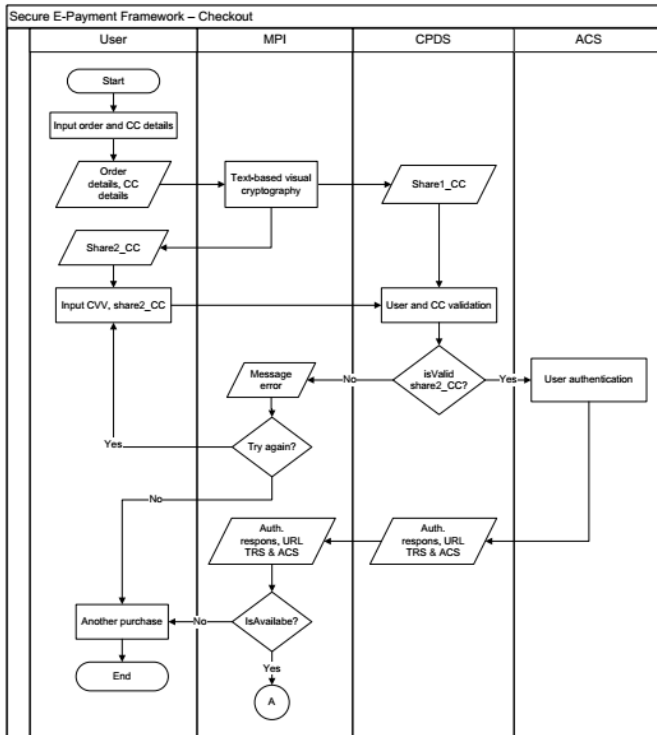


Fig. 4. Checkout flowchart

In phase in Figure 4, user enters credit card details, such as the cardholder's name, credit card number, and expiration date. No need to enter CVV/CVC on this page. After the user submits the inputs, the MPI then generates two shares using visual cryptography from credit card details, transaction id, and time stamp. One is sent to the user, and the other is sent to the CPDS. The user then redirected to the CPDS.

In the CPDS, user gives his/her share and 3 digits of CVV/CVC. The CPDS also sends its corresponding share that was received from the MPI. By stacking these two shares, the credit card details are appeared so that the CPDS can verifies the user's credit card and forwards the request to the corresponding card issuer ACS to determine whether authentication (or proof of an attempted authentication) is available for the credit card. The response from ACS is returned to the MPI includes the URL of card provider's TRS and the card issuer ACS to which the merchant will send the purchase authentication request.

In the checkout process in Figure 5, the user interacts with the MPI, Transaction Routing Service (TRS), and ACS. All of the authentication requests and responses are processed via the card provider's TRS to ensure a timely response to the merchants. After receiving authentication response from the ACS, MPI then checks whether the authentication for the

credit card is available or not. If authentication is available, MPI sends the purchase authentication request to the ACS via the card provider's TRS via the user's browser using the URL received in previous step. That request contains information about purchase transaction, including detailed merchant information and the URL of the merchant.

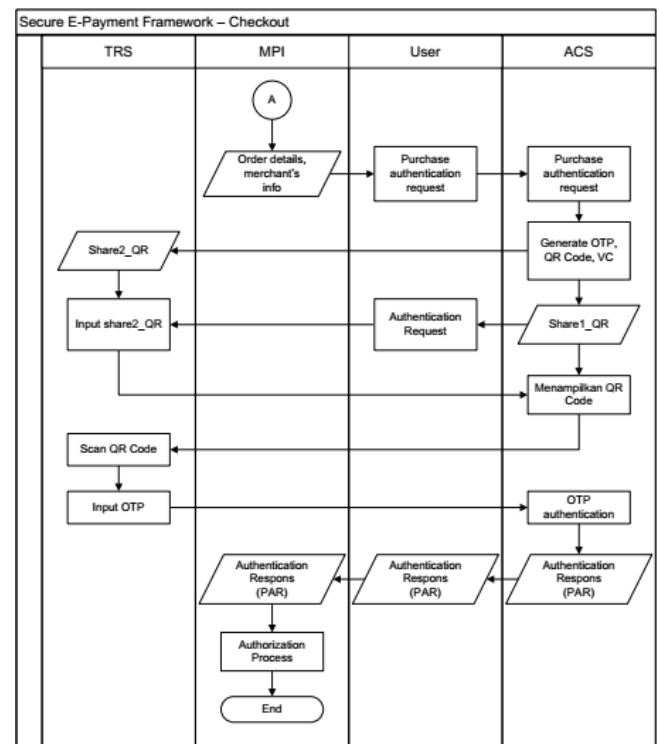


Fig. 5. Checkout flowchart (cont.)

ACS then formats an authentication request for the user. The authentication request is in form of a QR Code containing OTP. ACS generated two shares from QR Code using visual cryptography. One share is temporary stored by the ACS, and the other is sent to the user via email. Authentication request is returned via TRS to the user's browser. In the browser window, during authentication request, the user sees the following: after submitting CVV/CVC and share in previous step, the user sees a window that contains purchase details and that prompts the user to submit his/her share.

After the user submits his/her share, ACS then determines whether the share submitted by the user is valid or not. ACS combines the user's share with its share. If both shares are valid, the QR Code will appear in the browser window. Then the user scans that QR Code and gets the OTP, and sends it. If OTP is valid, ACS formats a Payer Authentication Request (PAR) with appropriate values, including user authentication status, and then returns PAR to the TRS which forwards the response to the MPI via the user's browser.

MPI validates the PAR and passes the results of the authentication attempt to the merchant server. Based on the data received from the MPI, the merchant server determines whether to proceed with authorization, as shown in Figure 6.

If the merchant advises the MPI that authentication failed, the merchant should requests another form of payment from the user. If authorization is appropriate, the merchant server sends an authorization request to the merchant's acquirer. The acquirer sends the authorization request to the card issuer via card provider network. The card issuer receives and processes

the authorization request a returns an authorization response. The issuer may choose to approve or to decline the authorization request for reasons unrelated to the credit card authentication, e.g. insufficient funds, closed account, etc. If the card issuer authorizes the transaction, the merchant will displays an order confirmation as usual. Until this step, payment is complete.

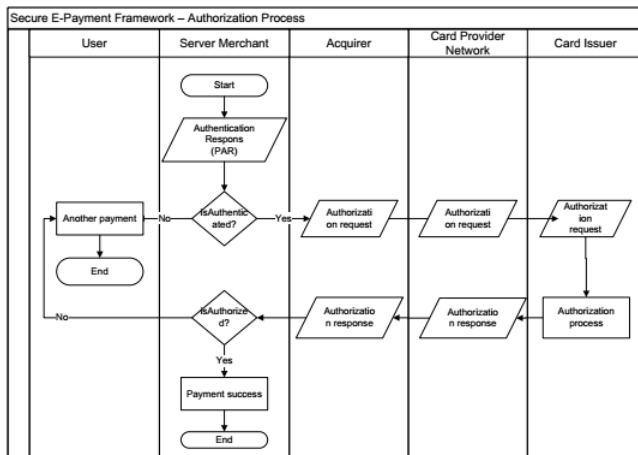


Fig. 6. Authorization process flowchart

V. IMPLEMENTATION

A. Registration Phase Algorithm

Phase	Registration
Initial State	Username, password, email, phone, captcha, etc.
Final State	Registration success, shares of captcha
Algorithm	IF isValid (username, password, email, phone, etc.) and isTrue (captcha) THEN Registration_success SEND share1_captcha to user SAVE share2_captcha to database ELSE Registration_failed

In the registration phase, there are two processes: generation of random captcha image and visual cryptography to split the captcha image into two shares, *share1_captcha* and *share2_captcha*.

B. Login Phase Algorithm

Phase	Login
Initial State	Username, Password
Final State	Username is valid, password is valid, username is invalid, password is invalid
Algorithm	IF isTrue (username, password) and isValid(captcha) THEN Username_valid ELSE Username_invalid, Password_invalid

In the login process, in addition to checking username and password, the system also checks the *share2_captcha* submitted by the user. *Share2_captcha* is stacked with *share1_captcha* from the merchant's database. If the captcha string is visible, visually it can be said that the user is valid. The user is then asked to enter the captcha string that is visible on screen. If the user's share is invalid, the captcha string will not be visible and the user will not be able to login to the merchant's site. Figure 7 shows an example of captcha shares.

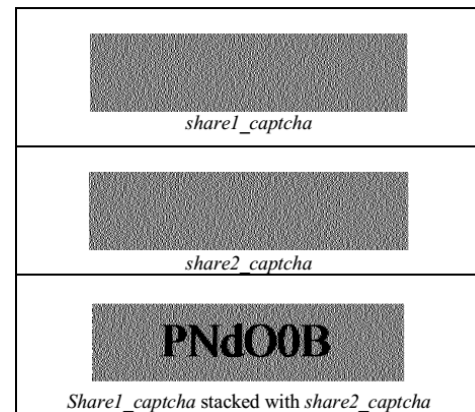


Fig. 7. Captcha shares

C. Checkout Phase Algorithm

Phase	Checkout
Initial State	Order_id, User_id, CC_information
Final State	Payment_success, Payment_failed
Algorithm	IF isValid (user) and isEnough (CC limit) THEN Payment_success ELSE Payment_failed

In the checkout phase, the user is asked to fill in the payment details and submits the credit card details (card number, name on card, expiration date). The MPI performs text-based visual cryptography using scheme (2, 2) with pixel replacement, resulting two shares [2].

Algorithm 1. Text to image shares encryption

Input: secret text file

Output: Image shares

1. Read the text containing the credit card information
2. Initialize the 2 random noisy image shares (*share1_CC*, *share2_CC*) in PNG format
3. Move all the information into buffer
4. Calculate the height and weight of the image shares
5. Width = no. of characters in a line
6. Height = no. of lines in a secret text file
7. Read each line from the contents in a buffer
8. Begin
9. Select each character from the line
10. Compute ASCII value of each character
11. Calculate the individual pixel using SetRGB() in C#
12. Place every pixel of a line in *share1_CC* and *share2_CC* alternately
13. Store pixel on the image shares based on its x, y coordinates which denote the position of the character in a line and line number respectively
14. Repeat this process till the end of file
15. End
16. Finally, save the image shares

Figure 8 shows an example of credit card details shares.

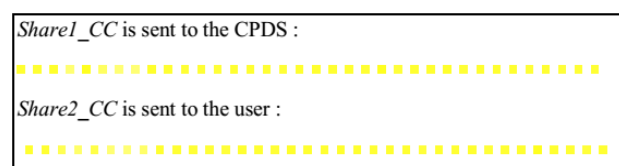


Fig. 8. Credit card details shares

The MPI then redirects the user to the CPDS, at the same time sending purchase details, merchant's details (account, URL), and share1_CC. Figure 9 shows a credit card validation window.

Credit Card Validation

Please validate your Credit Card.

CVC/CW:

CC Share: Payment102...areCC1.png

Fig. 9. Credit card validation window

On the CPDS, user gives his/her CVV/CVC and share2_CC sent by the MPI. If share1_CC and share2_CC are valid then the CPDS can read the credit card details using following algorithm.

Algorithm 2. Image shares to text decryption
Input: image shares
Output: secret text
1. Read the random noisy image shares
2. Initialize the string buffer
3. Select each and every pixel from the line using getRGB() in C#
4. Find ASCII value of the selected pixel
5. Find appropriate character from ASCII value and place it in a buffer
6. Rewrite all the data from the buffer into secret text
7. Finally, read the information in the secret text

The CPDS then forwards the request to the corresponding card issuer ACS. The response from ACS is returned to MPI. If authentication is available, MPI sends the purchase authentication request to the ACS. ACS then formats an authentication request for the user in form of a QR Code containing OTP. ACS generated two shares from QR Code using visual cryptography using (2, 2) scheme with 4-subpixels expansion. Figure 10 shows an example of QR Code generated by the ACS and QR Code resulted from stacking two shares.

User then scans the QR Code and sends OTP to card issuer as a payment authorization. If the OTP is valid, then the ACS will return the message that the authentication process was successful, formats a Payer Authentication Request (PAR), and returns it to the MPI to be forwarded to the merchant's server.

Merchant's server then processes the authorization by sending an authorization request to the corresponding acquirer. Acquirer then forwards the authorization request to the card issuer via card provider network. Card issuer receives and processes the authorization request, and then returns the authorization response. Card issuer may accept or reject the authorization request for a reason that is not related to the authentication result, e.g. insufficient fund, inactive account, etc. If the card issuer authorizes the transaction then the payment is successful.



Fig. 10. Original QR Code and QR Code resulted from stacking two shares

The purpose of phishing is to gain user's credentials, such as credit card details. When the user receives a fake email asking for credit card data confirmation by clicking on a link that goes to a site that resembles the original site, the user needs to submit his/her captcha share. If after that submission the captcha string doesn't appear, then the user doesn't need to give his/her credit card details, because it can be ascertained that the site is fake. This scenario should never happen because the merchant is not allowed to store user's credit card details, so there is no need to confirm that data.

If in such a way the phisher or attacker has gained the user's credit card details and use them to make a transaction on the merchant's site, to be able to make a successfully credit card payment transaction, the phisher or attacker still needs to have an access to the user's email and phone number registered on the card issuer associated with the credit card account.

The user is allowed to make a purchase without registering toward the merchant's site. It is possible because the data needed to confirm the credit card payment are only the data related to the credit card account, which are stored in the database of card provider and card issuer's server, and that data are not directly related to user's registration data on the merchant's site.

A. Pros

- Payment authentication is gained because only legitimate participants have share images in every transaction.
 - Captcha shares – only owned by the legitimate user and merchant. Attacker will not be able to access the user's account even if he has username and password, because he does not have a captcha share that was generated in the first time the user registering to the merchant's site.
 - Credit card shares – only owned by the legitimate user and the card provider. Attacker will not be able to gain anything even if he has successfully gotten the credit card's share, because there was a validation process by entering the CVV/CVC in the card provider's portal.
 - QR Code shares – only owned by the legitimate user and merchant.

- Payment authorization is gained by the presence of the OTP stored in the QR Code, which will be requested every time a credit card payment transaction occurs. This will be minimized the possibility of the transactions carried out by unauthorized user.
- User's confidentiality is gained by a visual cryptographic process that hides the credit card details during data communication between networks. Besides that, there is no credit card data stored in the merchant's database, so it is safe from the risk of data misuse and/or data theft, both internal (merchants) or external.
- Data integrity is more assured because the data is encrypted using visual cryptography so it can't be changed by unauthorized user.

B. Cons

- Payment transaction path become longer and more complicated so it may reduce the user's comfortability.
- In every transaction, user must submit the credit card details in the merchant's site and the CVV/CVC in the card provider's portal.
- User must keep his/her share properly, because if the share is damaged or lost, the user will not be able to login to the merchant's site or recover it.
- The merchant's server, the card provider's server, and the card issuer's server get busier because of the encryption process (merchant, card issuer), credit card validation (card provider), also the generation of the OTP and QR Code (card issuer).
- The user must have at least two devices, the one is used for purchase transaction at the merchant's site and the other for scan the QR Code.

CONCLUSIONS

Nowadays, phishing attack are as common as captures and store the user's secret information. The proposed method is based on using visual cryptography. Visual cryptography is applied to captcha image in registration phase, credit card details and QR Code containing OTP in purchase and payment

phase. The test results show that this proposed method can prevent phishing and identity theft, in sense that the authentication, authorization, and confidentiality are gained. Phishing can be prevented because only the legitimate user and the legitimate merchant having the captcha shares. Credit card payment transaction by unauthorized user can be prevented because only the legitimate can confirm the payment using the OTP stored in the QR Code, where the share of the QR Code will only be sent to user's email integrated with the credit card account. Credit card data theft can be prevented because that data is not stored in the merchant's database.

REFERENCES

- [1] D. James and M. Philip, "A Novel Anti-Phishing Framework Based on Visual Cryptography", International Journal of Distributed and Parallel Systems, Vol. 3, No.1, January 2012
- [2] K. Brindha and N. Jeyanthi, "Securing Portable Document Format File Using Extended Visual Cryptography to Protect Data Storage", in International Journal of Network Security, Vol. 19, No. 5, September 2017.
- [3] M. Naor and A. Shamir, "Visual Cryptography", in Proc. EUROCRYPT, 1994.
- [4] N. Chaudari and P. Parate, "Secure Online Payment System using Visual Cryptography", in International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 2, February 2016.
- [5] N.R. Jain, K. Ujwal, S. Apsara, P. Nikhil, and D. Tejashri, "Advance Phishing Detection using Visual Cryptography and One Time Password", in International Journal of Advanced Research in Science, Engineering and Technology, Vol. 3, Issue. 4, April 2016.
- [6] S. Akolkar, Y. Kokulwar, A. Neharkar, and D. Pawar, "Secure Payment System using Steganography and Visual Cryptography", in International Journal of Computing and Technology, Vol. 3, Issue 1, January 2016.
- [7] S. Roy and P. Venkateswaran, "Online Payment System using Steganography and Visual Cryptography", in IEEE Students Conference on Electrical, Electronics and Computer Science, April 2014.
- [8] VISA, "Verified by Visa: Acquirer and merchant implementation guide", U.S. Region, 2011.
- [9] W.P. Fang, J.H. Hsu, and W. Cheng, "Text-based Visual Secret Sharing", in International Journal of Computer Science and Network Security, Vol. 13, No. 5, May 2013.
- [10] X. Cao, L. Feng, P. Cao, and J. Hu, "Secure QR Code Scheme Based on Visual Cryptography", on 2nd International Conference on Artificial Intelligence and Industrial Engineering, Vol. 133, January 2016.