International Journal of Information and Communication Sciences
2018; 3(3): 104-109
http://www.sciencepublishinggroup.com/j/ijics
doi: 10.11648/j.ijics.20180303.14
ISSN: 2575-1700 (Print); ISSN: 2575-1719 (Online)

Pixel Value Graphical Password Scheme: Fake *Passpix* Attempt on Hexadecimal Password Style

Mohd 'Afizi Bin Mohd Shukran, Mohd Sidek Fadhil Bin Mohd Yunus

Faculty of Defence Science and Technology, Universiti Pertahanan Nasional Malaysia, Kuala Lumpur, Malaysia

Email address:
afizi@upnm.edu.my (M. 'A. B. M. Shukran), sidek.my@gmail.com (M. S. F. B. M. Yunus)

To cite this article:
Mohd 'Afizi Bin Mohd Shukran, Mohd Sidek Fadhil Bin Mohd Yunus. Pixel Value Graphical Password Scheme: Fake *Passpix* Attempt on Hexadecimal Password Style. *International Journal of Information and Communication Sciences*. Vol. 3, No. 3, 2018, pp. 104-109. doi: 10.11648/j.ijics.20180303.14

Received: November 27, 2018; **Accepted**: December 5, 2018; **Published**: December 6, 2018

Abstract: Early design of graphical password scheme solved the strong alphanumeric password deficiencies in term of hard to memorized, but they was exposed to password shoulder surfing and complicated to produced even by the legitimate users. Motivated to solve the problems, pixel value graphical password scheme was developed in 2012 where it require user to load any their desired digital image as their password. The term *PassPix* is referring to the pixel value that extracted from the loaded image file derived from the words password pixel. Pixel value in the form of RGB color strength could be representing as 3-octets 8-bits color code (0 to 255) or Hexadecimal code (000000 to FFFFFF) produced 16,777,216 combinations. Pixel value graphical password increases the password strength by implementing two-dimensional 8 by 8 grid extraction that increases the password strength to 1,073,741,824 combinations. It would result the password could be long or super long. Currently, pixel value graphical password scheme using the 8-bits color code as the pixel value and password that stored into database which could produce up to 576 characters. This would be require more storage capacity to store the password that brings into a suggestion to extracting the pixel value using hexadecimal code which produced 384 characters only. Further study on that idea it is found that the number of password characters did not brings a significant impact on-disk data size and it is easy information to construct a fake *PassPix* which is discussed further in this paper. This paper is organized into 5 sections where the background and concept of pixel value graphical password scheme is introduced in section 1. Section 2 of this paper is describing the password style of pixel value graphical password. The fake *PassPix* reconstruction is briefly explained in section 3 and section 4 is discussed further about the fake *PassPix* findings. The conclusion of this study can be found in section 5 and at the end of this paper, there are list of several references used in this study. Hopefully this paper will brings major contribution for safer pixel value graphical password scheme.

Keywords: Pixel Value, Graphical Password, *PassPix*, RGB Color Scheme, 8-Bits Color Code, Hexadecimal Color Code

1. Introduction

In 2012, pixel value graphical password scheme [1] was introduce to solve the previously developed graphical password complexity and password vulnerability. Before its development, most of the graphical password scheme was involving click on image password such as Blonder's method [2] and *PassPoint* method [3]. By the end of 2010, click-based method was implemented widely as the additional log in security media [4]. There are also a few developments on draw-based graphical password such as signature drawing [5] and *PassGo* method [6]. Some of the developed graphical password scheme is complex as a method by Sobrado & Birget [7] and some easy to duplicate [8]. Pixel value graphical password scheme was introduced to reduce the user complexity and eliminate the password duplication issue.

With pixel value graphical password scheme, users only need to pick any of their personal desired images and load it to authentication system as their *PassPix*, the term for password pixel. It is far simpler for users to create and authenticate their password using this method [9] compares to the other graphical password scheme where user is freely to pick a password according to their personal preferences and they loved to memorizing it. To create a password or log in to pixel value graphical authentication system, users are requiring providing their username as their unique ID and loading their desired image as *PassPix*. The authentication interface is illustrated in figure 1.

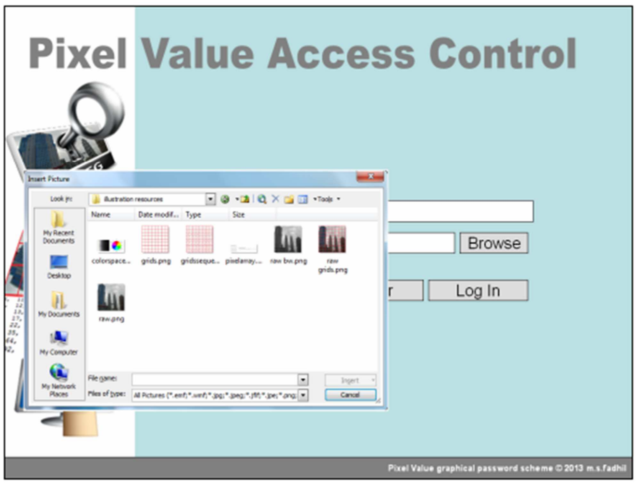

Pixel Value Access Control
Browse
Log In
Pixel Value graphical password scheme © 2013 m.s.fadhil

Figure 1. *Pixel Value access control interface.*

By allowing users to pick their image freely as their desired, users would be easily memorizing it password easily. Unlike the other graphical password, the loaded image by users are not disclosed to the other user which the password information or clue is safe. In addition, when users choose an image from their personal storage media, the *PassPix* is only exclusively accessible by the legitimate user.

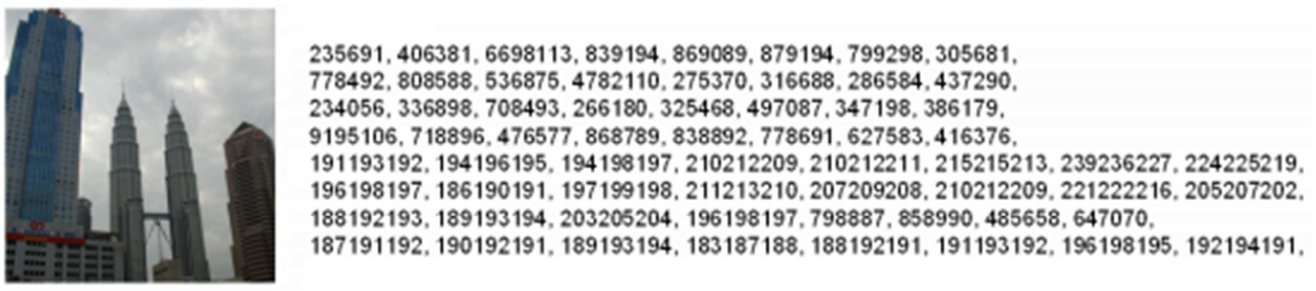
2. The Password Style

For, pixel value graphical password scheme, *PassPix* in the form of digital image file, is the main component that carries the password information. The password will be extracted by pixel value authentication system to produce the image pixel value. The loaded image file is divided into logical two-dimensional grids and pixel value from each grids being extracted. Example of pixel value in 8-bits code and hexadecimal code 1s illustrate in following table 1.

Table 1. *The 8 bits and hexadecimal pixel value color codes.*

Color name	Color	8 Bits Code	Hexadecimal
Black		000	FFFFFF
Red		25500	02550
Green		02550	00FF00
Blue		00255	0000FF
White		255255255	000000

Whole pixel value is arranged as one single text to be stored into the user database. The pixel value is in the form of three octets RGB color codes which could be the 8 bits color code or hexadecimal code [10]. It is a super long password and meaningless which make it a strong password. Example of an extracted 8 bits color code pixel value from an image using 8 by 8 grids is shown in figure 2.



```
235691, 406381, 6698113, 839194, 869089, 879194, 799298, 305681,
778492, 808588, 536875, 4782110, 275370, 316688, 286584, 437290,
234056, 336898, 708493, 266180, 325468, 497087, 347198, 386179,
9195106, 718896, 476577, 868789, 838892, 778691, 627583, 416376,
191193192, 194196195, 194198197, 210212209, 210212211, 215215213, 239236227, 224225219,
196198197, 186190191, 197199198, 211213210, 207209208, 210212209, 221222216, 205207202,
188192193, 189193194, 203205204, 196198197, 798887, 858990, 485658, 647070,
187191192, 190192191, 189193194, 183187188, 188192191, 191193192, 196198195, 192194191,
```

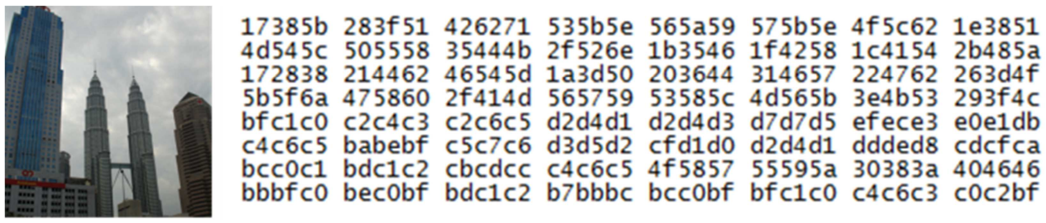
Figure 2. *8-bits color code password output.*

Output password:
2356914063816698113839194869089879194799298305681778492808588536875478211027537031668828658443729023405633689870849326618032546
8497087347198386179919510671889647657786878983889277869162758341637619119319219419619519419819721021220921021221121521521323923
6227224225219196198197186190191197199198211213210207209208210212209221222216205207202188192193189193194203205204196198197798887
858990485658647070187191192190192191189193194183187188188192191191193192196198195192194191

When the password is arranged into a single text, it would form a 192 characters password and up to 576 characters password. The non-fixed and mash number of characters in 8-bits color code, from 000 to 255255255, would result the number of grids and the color code combination is hard to identify for further fake PassPix reconstruction. However, when the pixel value is extracted in the form of hexadecimal codes, it is otherwise where the hexadecimal code is always fixed 6 characters.

3. Fake *PassPix* Reconstruction

Even though both hexadecimal and 8 bits color code produced 1,073,741,824 password combination using 64 grids (8 by 8 grids), the hexadecimal text structure is fix to 6 characters. Unlike the 8 bits color code, minimum character for each grid is 3 character and up to 9 character which it is perfect for confusing the color combination. Example of the hexadecimal password output is shown in figure 3.



```
17385b 283f51 426271 535b5e 565a59 575b5e 4f5c62 1e3851
4d545c 505558 35444b 2f526e 1b3546 1f4258 1c4154 2b485a
172838 214462 46545d 1a3d50 203644 314657 224762 263d4f
5b5f6a 475860 2f414d 565759 53585c 4d565b 3e4b53 293f4c
bfc1c0 c2c4c3 c2c6c5 d2d4d1 d2d4d3 d7d7d5 efece3 e0e1db
c4c6c5 babebf c5c7c6 d3d5d2 cfd1d0 d2d4d1 dded8 cdcfca
bcc0c1 bdc1c2 cbcdcc c4c6c5 4f5857 55595a 30383a 404646
bbbfc0 bec0bf bdc1c2 b7bbbc bcc0bf bfc1c0 c4c6c3 c0c2bf
```

Figure 3. *The Hexadecimal color code password output.*

Output password:
17385b283f51426271535b5e565a59575b5e4f5c621e38514d545c50555835444b2f526e1b35461f42581c41542b485a172838214462465 45d1a3d502036443146
57224762263d4f5b5f6a4758602f414d56575953585c4d565b3e4b53293f4cbfc1c0c2c4c3c2c6c5d2d4d1d2d4d3d7d7d5efece3e0e1dbc4c6c5babebfc5c7c6d3d5
d2cfd1d0d2d4d1ddded8cdcfcabcc0c1bdc1c2cbcdccc4c6c54f585755595a30383a404646bbbfc0bec0bfbdc1c2b7bbbcbcc0bfbfc1c0c4c6c3c0c2bf

Not just it produced shorter password space, the code is guessable. Since hexadecimal code is always 6 characters, the information about number of grids and the pixel value for each grids is exposed. Assuming there is no information about the grid dimension used, by counting 6 characters to form a block from that super long password, the total number of grids applied for this password can be found. To form a two-dimensional grid, simply just square root the total number of counted blocks and arrange the blocks accordingly.

For an example:

Output Password:

17385b283f51426271535b5e565a59……α

Divide text into block of 6 characters:

17385b 283f51 426271 535b5e 565a59 ……α

Count the number of blocks:

[1]	[2]	[3]	[4]	[5]	[n] ……α
17385b	283f51	426271	535b5e	565a59	

Calculate the dimension:

$\sqrt{n} = Y \times \varkappa$

(example based on figure 3: 64 blocks, $\sqrt{64} = 8 \times 8$ dimension)

Where; α = last character
n = sum of blocks
Y = size of Y axis
$\varkappa$ = size of X axis

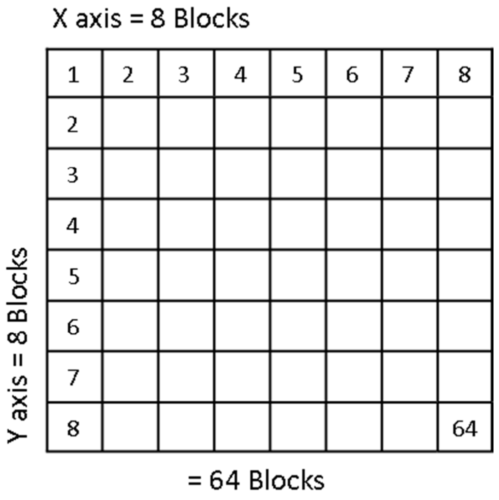
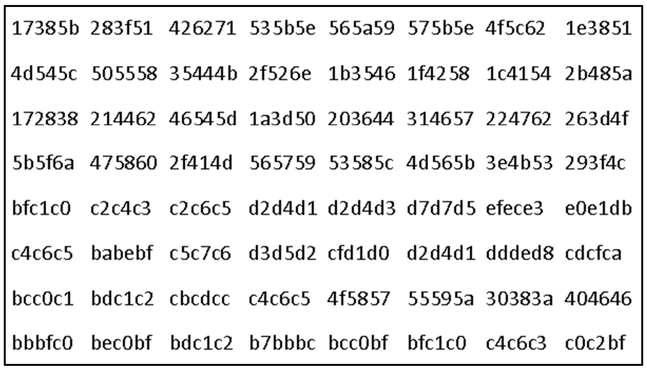


Figure 4. *Two-dimensional grids reconstruction.*

From the example, the total blocks count is used as the information to find the axis size where square root of 64 blocks is 8. It is mean that both dimension axis, X and Y, have the size of 8 blocks. When the size of X axis and Y axis is known, the two-dimensional grids can be constructed as illustrated in figure 4.

The hexadecimal code from each block acquired from the division step previously is used to fill in each block with color accordingly for each block. The arranged hexadecimal code block is illustrated in figure 5 and the colored fake *PassPix* is shown in figure 6.



17385b	283f51	426271	535b5e	565a59	575b5e	4f5c62	1e3851
4d545c	505558	35444b	2f526e	1b3546	1f4258	1c4154	2b485a
172838	214462	46545d	1a3d50	203644	314657	224762	263d4f
5b5f6a	475860	2f414d	565759	53585c	4d565b	3e4b53	293f4c
bfc1c0	c2c4c3	c2c6c5	d2d4d1	d2d4d3	d7d7d5	efece3	e0e1db
c4c6c5	babebf	c5c7c6	d3d5d2	cfd1d0	d2d4d1	ddded8	cdcfca
bcc0c1	bdc1c2	cbcdcc	c4c6c5	4f5857	55595a	30383a	404646
bbbfc0	bec0bf	bdc1c2	b7bbbc	bcc0bf	bfc1c0	c4c6c3	c0c2bf

Figure 5. *Hexadecimal code arrange into 64 blocks.*

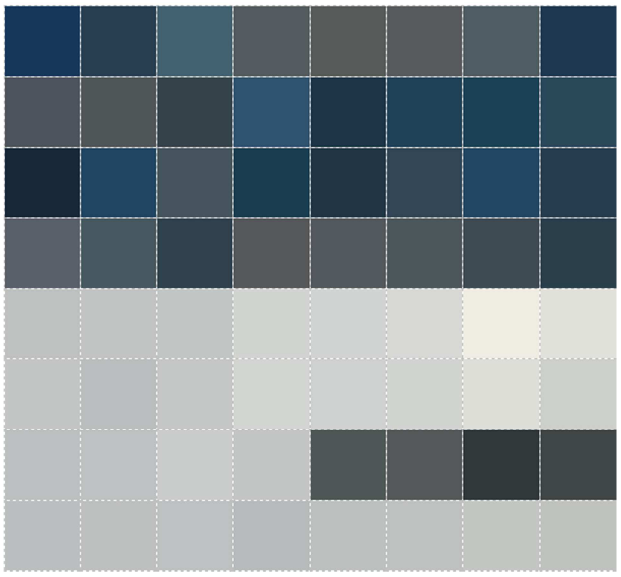
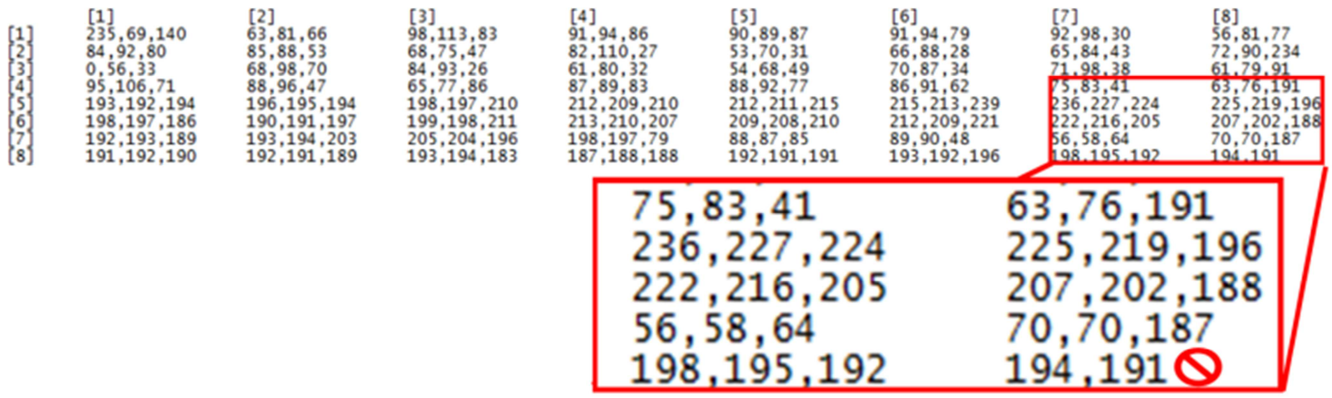


Figure 6. *The grids color filling based on hexadecimal code.*

The fake *PassPix* is build using html table with 8 columns and 8 rows that resemblance to the two-dimensional 64 grids as illustrated in figure 4. Every block were set 30 pixel height and 30 pixel wide with background color for each blocks as arrange in figure 6. The html file then open in browser and the colored table is capture with screen capture. The production of fake *PassPix* was done in the same machine where the pixel value access control system is host and the fake *PassPix* was tested locally. At first, the original *PassPix* was used to ensure that there is no system error before the fake *PassPix* being loaded. As expected, the *PassPix* was able log in as a legitimate user and log out before starting the new session.

Using the same registered user account in the database, this fake *PassPix* block however, fails to gain its authentication on pixel value authentication system which discussed further next section. Before that, lets take a look on the possibility for 8-bits color code fake *PassPix* reconstruction.

As the 6 characters rule in a block for hexadecimal code, the 8-bits color code also has a term to be fulfilled to form a block where each block containing 3 octet numbers and each octet value are from 0 to no more than 255. This mean, each block could containing only 3 numbers and up to 9 numbers. By using the same technique but different block rule, the password output is divided into blocks as illustrate in following figure 7.



```
     [1]          [2]          [3]          [4]          [5]          [6]          [7]          [8]
[1]  235,69,140   63,81,66     98,113,83    91,94,86     90,89,87     91,94,79     92,98,30     56,81,77
[2]  84,92,80     85,88,53     68,75,47     82,110,27    53,70,31     66,88,28     65,84,43     72,90,234
[3]  0,56,33      68,98,70     84,93,26     61,80,32     54,68,49     70,87,34     71,98,38     61,79,91
[4]  95,106,71    88,96,47     65,77,86     87,89,83     88,92,77     86,91,62     75,83,41     63,76,191
[5]  193,192,194  196,195,194  198,197,210  212,209,210  212,211,215  215,213,239  236,227,224  225,219,196
[6]  198,197,186  190,191,197  199,198,211  213,210,207  209,208,210  212,209,221  222,216,205  207,202,188
[7]  192,193,189  193,194,203  205,204,196  198,197,79   88,87,85     89,90,48     56,58,64     70,70,187
[8]  191,192,190  192,191,189  193,194,183  187,188,188  192,191,191  193,192,196  198,195,192  194,191
```

```
75,83,41        63,76,191
236,227,224     225,219,196
222,216,205     207,202,188
56,58,64        70,70,187
198,195,192     194,191
```

Figure 7. *8-bits code arrange into 64 blocks.*

The 8-bits code was successfully arrange into 64 blocks style just like the hexadecimal code but only the last block missing the third octet. When the third octet value become unknown, a RGB color in the block 64 cannot be form and the password was incomplete. Comparison between figure 2 and figure 7, there are few block that containing different value than the original value such as the first block and the second block that caused the last block missing its third octet where the last octet of the first block absorbing the first octet of the second block value.

This show that the key benefit of using the 8-bits code as the password text for pixel value graphical password is the method of confusing for fake *PassPix* reconstruction. The number of characters in each octets are not fix where it could become only one character and up to three numbers. For a block, there would be three numbers and up to nine numbers. There are also possibilities where the text becomes more confusing that would cause more octets missing or even blocks missing. The missing numbers will be remain null and could not be replace with number 0 where it produce different pixel value and the password text will be completely wrong and deny.

4. Findings and Discussions

A dynamic analysis on pixel value graphical password scheme in 2014 [11] conclude that, there are several factors for duplicated *PassPix* fail to authenticate others than the pixel value itself. Unlike the fake *PassPix* produced in this study, the specimen used in that testing was originally from the actual *PassPix*. One of the denial factors is the *PassPix* resolution where the pixel distribution difference will produced different pixel value. The different between the original *PassPix* and the fake *PassPix* is illustrated in figure 8 below.

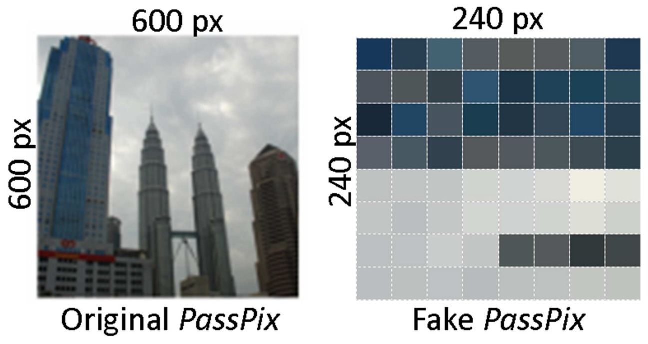


Figure 8. *The difference of image resolution between original PassPix (left) and fake PassPix (right).*

In the analysis study as mentioned before, the researcher explained that a small resolution digital image has gross pixel on shape edge compared to a high resolution digital image which has a sharper pixel on image edge. That makes the color on small resolution digital image scatter more on to surrounding area which affected the pixel value. In the case of shapeless image such as the fake *PassPix*, the area affected would be on the edge of each block border with the possibilities that the color on the edge is mixing among two block color and affecting the pixel value for each block.

For the second attempt, it has been decided to increase the resolution by 5 times which produce 1,200 pixels by 1,200 pixels image or 2 times larger than the original *PassPix* by assuming that the edge of block border are sharper than previous 240 pixels by 240 pixels fake *PassPix*. However, as previous fake *PassPix*, this large fake *PassPix* also failed to authenticate by pixel value access control system.

This study assumes there are two possibilities for that unsuccessful attempt. First, back to previous argument on image resolution that pixels compositions are different from the original *PassPix*. However, the reason for the reconstruction of larger resolution fake *PassPix* to get a sharper pixel edge on the border than previous fake *PassPix*. Then this study focus deeper on the second possibility that on the fake *PassPix* there are existent of several block gap, a blank area between block. It might be unnoticeable on small resolution fake *PassPix* and become obvious on large resolution fake *PassPix*. The block gap is shown on figure 9.

Even though that the table border size was set to 0 pixel and no border-color set, the block gap still exist and that make us to decide to resize that fake *PassPix* for 50% less than current resolution or same as the 600 pixels by 600 pixel original *PassPix*. Then an image editing software is used in the operating system of the host machine. This study assumes that effort will shrink the block gap on the fake *PassPix* and it could become unnoticeable for pixel value access control system. However, as the previous attempt, this attempt also fails to authenticate that fake *PassPix*. Based on this third attempt, four assumptions have been made about the fail factor of this attempt as listed in the following table 2.

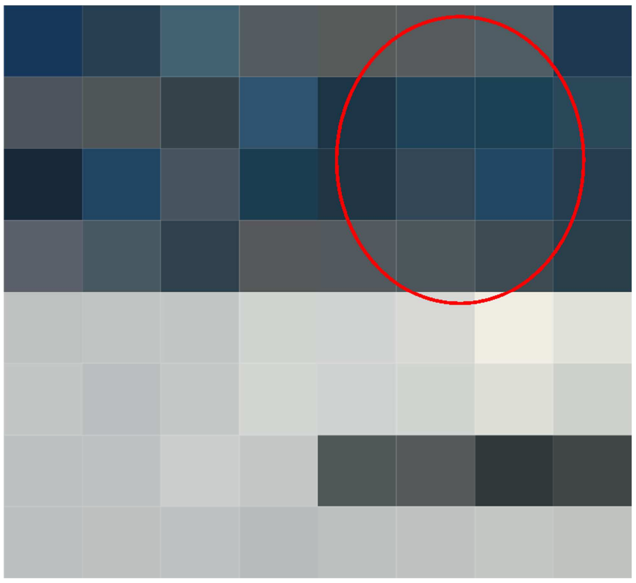


Figure 9. *The block gap on fake PassPix in the rounded area and several other areas.*

Table 2. *Fake PassPix fail factors assumption.*

Factors	Descriptions
Block Gap	Shrunk but still exist and affected the pixel value of a block.
Graphic editing software colour scheme	The simple software might be fail to identify and recreate the exact colour code of each block in fake *PassPix*.
Screen capture function	As the limitation of graphic editing software, the screen capture function colour setting might be the contributing factors.
Fake *PassPix* resolution issue	This issue might be a tiny possibility for this attempt and require further research.

Ahead of this attempt, this study are not conducting further attempt due to tools limitation (such as image generator and editor software) and graphical skills. Moreover, issues on graphical tools and skills are not emphasis in this study.

As the fake *PassPix* did not successfully pass this attempt, it is assumed that graphical tools or software plays vital role for future attempt. For recap, the fake *PassPix* was generated using generic html table script and displayed on common web browser. The fake *PassPix* block being capture using screen capture or print screen feature and paste it on built in graphic editor application. The block on graphic editor software canvas is saved as a digital image file.

The screen capture process itself might be the cause of the fake *PassPix* is not successfully authenticated. The capture image needs to be crop precisely on the edge of outer border to prevent additional pixel in the fake *PassPix* that cause pixel value wrongly interpret. In contra, when the image being cropped smaller than the actual size, the border would not logically divide equally and cause the pixel value for blocks would be different.

Above of all, in this study, the blocks were generated using a generic html table code with 0 pixel border thickness. With common web browser as the media to display the blocks, the table border might be the cause of the block gap as shown in figure 9 before. In this case, even though the html method is the easiest way to construct the 64 blocks image, it is not a perfect method for constructing the blocks. It might be required more graphical specific tools for blocks generator.

5. Conclusion

This study is not promoting and encouraging the act of breaking the password especially for pixel value graphical password scheme. As the graphical password scheme researches are just 22 years old, it is require more and more research and development effort in order to make it reach the mature state as the alphanumeric password. However, the maturity of alphanumeric password does not guarantee the safety and security of the password information and those made the 6 years old pixel value graphical password scheme much safer and convenient password method to be implement.

The password style of pixel value graphical password scheme is reside in the digital image that called as *PassPix* which consisted of the pixel value or RGB value of logical blocks built during the extraction process can be represent in 3 octets 8-bits codes (000 to 255255255) or hexadecimal code (000000 to FFFFFF). Both color code has implementation benefits where 8-bits codes is the method of confusing the pixel value blocks reconstruction and hexadecimal code has shorter text length to store into database.

Section 3 of this research has proved that the hexadecimal code is an easy target to interpret the dimension of *PassPix* grids and the pixel value that reside in each blocks. For 8-bits code, the text was successfully arranged into 64 blocks but the third octet of the last block is missing due to the pixel value confusion on the first block and the second block.

Even though the fake *PassPix* was successfully constructed based on hexadecimal code, it is still failed to bypass the pixel value access control system. The major factor of this failure is the limitation of graphical tools and skills to generate the fake *PassPix* digital image file. While the html table is proven not a best method, it is assumed that it will require a precise block building tools that would not producing a block gap as in figure 9 and the capability to produce the digital image file without involving the screen capture feature.

Despite of the failure of the fake *PassPix* method exhibited in this study, this study also have proved that it is hard and complicated effort to break the pixel value access control system. Security attempt researches such this study were highly welcome in order to make pixel value graphical password scheme is harden and trusted to become a security parameter for log in system.

References

[1] M. A. M. Shukran and M. S. F. M. Yunus, "Pixel Value Graphical Password Scheme" PI 2013003261 Malaysia, 2015

[2] G. E. Blonder, "Graphical password" 5,559,961 Washington D. C., United State of America, 2015

[3] S. Chiasson, A. Forget, R. Biddle, and P. C. van Oorschot (2009). "User interface design affects security: Patterns in click-based graphical passwords," International Journal of Information Security, vol 8, pp 387, 2015

[4] Malaysian Banking Berhad, "M2U new enhanced Security Features. Maybank 2 U," [Online] https://www.maybank2u.com.my.

[5] A. F. Syukri, E. Okamoto and M. Mambo, "A user identification system using signature written with mouse," Australasian Conference on Information Security and Privacy, vol. 13, pp. 403-414, July 2015. [Springer, Berlin, Heidelberg]

[6] H. Tao, "Pass-Go, a new graphical password scheme" Doctoral dissertation, University of Ottawa (Canada), 2017.

[7] L. Sobrado and J. C. Birget, "Graphical passwords," The Rutgers Scholar, an electronic Bulletin for undergraduate research, vol. 4, pp. 12-18, 2016.

[8] R. Biddle, S. Chiasson and P. C. Van Oorschot, "Graphical passwords: Learning from the first twelve years" ACM Computing Surveys (CSUR), vol. 44, p. 19, 2015

[9] M. A. M. Shukran, M. S. F. M. Yunus, W. S. S. Shariff, M. S. Ariffin and K. Maskat, "Pixel Value Graphical Password Scheme: Identifying Design Features and Requirements," Applied Mechanics and Materials, Vol. 548, pp. 1561-1565, 2014, Trans Tech Publications.

[10] "HTML Color Codes," [Online] https://htmlcolorcodes.com/, September 3, 2015.

[11] M. S. F. M. Yunus, "Dynamic Analysis on Pixel Value Graphical Password Scheme," Master dissertation, National Defense University of Malaysia, 2016.