Journal of Iraqi Al-Khwarizmi (JIKh) Volume:6 Issue:1 Year: 2022 pages: 12-17

# PassDirectory: An Algorithm to Hide Login Information in Smart Devices Images

Sameera Abbas Fadhel[1, a)] and Enas Ali Jameel [2, b)*]

[1,2]*Computer Sciences Department, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq*

[a)] *sameeraabbasfadhel@uomosul.edu.iq*

[b)]**corresponding author: EnasAliJameel@uomosul.edu.iq*

**Abstract.** Smart devices have proliferated in the past decade. These devices interact with all of our daily activities. These devices are filled with apps that collect data, save personal details, or suggest various activities to be followed. Most of these apps require users authenticate with passwords. This enforces users to create and remember multiple passwords or to use the same password for different applications. However, this makes it hard if not imposable to remember all created passwords. To address this issue, in this work, a new photo-masking algorithm is used to hide user-generated credentials in smartphones photos, named PassDirectory. The algorithm encrypts the login information using the simple Playfair algorithm. Subsequently, the encrypted data is embedded in saved photos in users' phones. To embed the data in the photos, two pixels are swapped to embed one data bit. The algorithm has been implemented and its data bit capacity, PSNR has been calculated. The algorithm's embedding capacity is smaller than other techniques, however, it is easy to implement in various devices without consuming power.

**Keywords.** PlayFair, Steganography, Smartphones, Passwords, Authentication

## INTRODUCTION

Devices all around the world to communicate with each other [1]. This communication requires data security and users' authentication. It has been reported that smartphones' users have on average 20 downloaded and installed applications [2]. These applications require login information, such as, usernames and passwords. In addition, these users are subscribes in hundreds of online websites and services that also require authentication. With this number of applications, websites and online services, users are required to memorize many usernames and passwords. It has been reported that users are required to remember over 25 passwords [3-5]. In addition, it has been reported that 50% of online users attempt to reuse their well-known passwords when creating new account for new online services [6]. This creates a security breach issue when one of these online service or website hacked different user's accounts on different servers and services will be under attack. This happened in Twitter breach, Yahoo, LinkedIn [7-9] and even gaming sites [10]. Another problem emerged that people do not change their credentials after a breach [11].

To tackle this issue, users are required to create a new credential for every online service and website. Nevertheless, users are attempting to reuse their old passwords with few changes. It has been reported that with a small guess that new password can be extracted from old ones [1]. Other sites and services attempted to tackle this issue by generating random hard and complex passwords for the users. This issue makes it harder for the users to memorize these passwords. To tackle this issue, in this work a new credential saving application for smartphones is proposed; named Pass Directory. The new application hides the credential using image steganography. Users' credentials will be encrypted utilizing PlayFair encryption algorithm. Subsequently, the encrypted data will be converted into their ASCII code. Each bit of the output data will be embedded into the photos saved in users' smartphone. Images' pixels will be swapped according to the value of each bit. This data will be embedded in different photos at the same time to reduce data losing if the user deleted the photos utilized for the embedding process. The algorithm has been written and the random data has been encrypted and embedded. Embedding capacity and visual impact of the images have been evaluated using PSNR.

The rest of this paper is organized as follows; the next section introduces some of the works that have been conducted in the area of password management system. Section 2 overviews the design system and how it operates. The experiment and the results is shown in section 3. We conclude this paper in section 4.

In **[13],** they developed Versipass, a password manager that incorporates key elements of password managers and cued graphical passwords to avoid existing problems of password memorability and associating passwords with accounts. Versipass also facilitates safe password reuse by allowing users to use the same image cue for multiple accounts. In **[14]** they presented the results of a study which aimed to identified which practices people do that they consider most important at protecting their security online. they compared self-reported security practices of non-experts to those of security experts. their findings show a discrepancy between the security practices that experts and non-experts report taking. In **[18]** the proposed method is inspired by Zhang and Wang's method and Sudoku solutions. the proposed method is more secure than Mielikainen's method and Zhang and Wang's method. From the experimental results, the visual quality of stego images produced by the proposed method is higher than 44 dB on average. In **[19],** A detailed comparative study is performed between the proposed approach and the other state-of-the-art approaches. This comparison is based on visualization to detect any degradation in stego image, difficulty of extracting the embedded data by any unauthorized viewer, Peak Signal-to-Noise Ratio (PSNR) of stego image, and the embedding algorithm CPU time. Experimental results show that the proposed approach is more secure compared with the other traditional approaches.

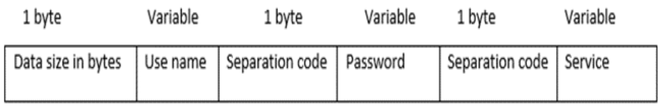
In this work, the proposed system adopts the locally saved passwords. However, to tackle the issues of this type, the system encrypts and hides the passwords in smartphones photos. In this way, even if the smartphone has been cracked, no one can reach the saved and hidden passwords in the smart device.

## PROPOSED SYSTEM OF PASS DIRECTORY

PassDirectory consists of four main parts; data encryption, embedding, data extraction and decryption. The following subsections describe these parts.

### *A.* Data Encryption

In this step, the system obtains users credential and the service name for these credentials. The system generates a data package consists of four parts; the size of the data in bytes, username, password and the service name. Figure 1 shows the structure of this package. We can observe that a one-byte separation has been added between the fields since the username and the password sizes are variable. The total size required for each data package equals the size of the username, password, service name, one byte for the size of the whole package and two bytes to separate these fields. After bundling the data in a package, the data is divided into segments of two bytes 'characters' in each segment. These segments are the input to the PlayFair encryption algorithm.



| 1 byte | Variable | 1 byte | Variable | 1 byte | Variable |
|---|---|---|---|---|---|
| Data size in bytes | Use name | Separation code | Password | Separation code | Service |

**FIGURE 1.** Package Format for each saved entry

To encrypt the segments, the system requires a password. This is the only password the user is required to memorize. This password is utilized to generate different PlayFair matrices. To start the encryption, the system gets the password from the user, deletes any duplicated characters in the password, creates a matrix or a table of 9x9 cells and fills the table with the unduplicated characters of the password. Subsequently, fills the remaining cells of the table with other characters from the table in figure 2. This means that the arrangement of the character in figure 2 table depends on the selected password. It worth mentioning that, the original PlayFair algorithm consists of tables of 5x5 with the small English characters only. The leveraged PlayFair in this work is an upgraded version that consists of characters 'capital and small', numbers and symbols.

After generating the 9x9 matrix, the table is used to substitute each character in the two characters' segments generated from the data package. The substitution follows the following roles

- If the two characters in the segment are in the same row in the generated matrix, replace each character with the character to its left and rotate to the left for border characters.
- If the two characters are in the same column, replace each character with the character under it in the same column and rotate down for border characters.
- If the two characters are in different rows and columns, draw a rectangle around these characters with each one of them in a corner of the rectangle. Subsequently, replace the character with the one in the opposite corner.
- If the segment has one duplicated character, add a separating character between them.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I |
| J | K | L | M | N | O | P | Q | R |
| S | T | U | V | W | X | Y | Z | 0 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| a | b | c | d | e | f | g | h | i |
| j | k | l | m | n | o | p | q | r |
| s | t | u | v | w | x | y | z | $ |
| @ | - | _ | + | / | { | } | < | > |
| ( | ) | ! | ^ | * | ‘ | = | . | ~ |

**FIGURE 2**. *9x9 PlayFair Matrix*

After repeating these steps for all segments, the encrypted data is ready to be embedded in images in step 2.

## *B.* Image Data Embedding ‘The second Step’

After encrypting the data and obtain the encrypted packages, the data is converted into its ASCII code. Subsequently, each bit is embedded in two pixels of a selected image, known as the cover image. The cover image will be utilized to hide the encrypted data by swapping the image’s pixels according to the following roles

- If the bit is zero, do not swap the pixels.
- If the bit is one, swap the pixel with its right pixel.
- In the third pixel change the least significant bit (LSB) into one if swapping occurred. If not converted to zero.

As an example, if the data bits are (1001), the first bit will swap the first and the second pixels of the image and the LSB of the third pixel will be 1. For the second bit, no swap occurs, however, the LSB of the sixth pixel will be zero. This means that these 4 bits requires 12 pixels to hide them. This means that a photo of 800x1000x3 pixels for RGB colored image can hide 100Kbyte of encrypted data. Moreover, this method allows the system to restore the original image from the cover image and deletes all the embedded encrypted data. Unlike the normal LSB hidden method, the hidden effect cannot be reversed. Moreover, the new method deals with colored images and hides data in all of its layers.

## *C.* Hidden Data Extraction ‘The Third Step’

After Hidden the data in the images, the system should have a way to extract them back. The extraction process is easy if the pixels are swapped, a data bit ‘1’ has been hidden. If no if no swap occurs, a data ‘0’ is hidden. However, how to know where are the data? Which smartphone’s images have been utilized to hide the data? To find the images with hidden data, the LSB of the first 50 pixels is utilized to save a preamble code. This code means that the image has been used to hide data. More bits or pixels mean less error for the data retrieving process. In such away, the system search for the code in the header of all images. If the code is found, the encrypted data packages are retrieved and the data package is reformatted for the decryption process.

## *D.* Data Decryption ‘The Last Step’

In this step, the formatted encrypted data retrieved from the third step is converted into characters using their ASCII code. Subsequently, these data is segmented into segments with two characters each. After that the 9x9 matrix of PlayFair algorithm is constructed with the password selected in the beginning. After the constructing of the matrix, the two characters’ segments are replaced with new characters from PlayFair matrix following these roles

- If the two characters in the segment are in the same row in the generated matrix, replace each character with the character to its right and rotate to the right for border characters.
- If the two characters are in the same column, replace each character with the character over it in the same column and rotate up for border characters.
- If the two characters are in different rows and columns, draw a rectangle around these characters with each one of them in a corner of the rectangle. Subsequently, replace the character with the one in the opposite corner.

Finally, the data package will be extracted.

## EXPEREMNET AND RESULTS

**FIGURE 3.** Images utilized in the experiment. On the left cover images without the embedded data on the right the cover image with the encrypted data embedded

Finally To evaluate the proposed credential management system , PassDirectory, the modified version of PlayFair with 9x9 matrix size has been written in Python. Moreover, a random number generator has been written to generate random numbers. Each two numbers generated are utilized to obtain a random character from a file with 100K lines of words. The extracted random characters have been grouped in variable sizes to generate usernames, passwords and services' names. These grouped data has been leveraged to generate data packages for the encryption process. PlayFair has utilized the 'PassDirectory' as a password for the encryption. The packages generated has utilized 'x' character as a code speeder between the username, password and service's name. A random binary number of 50 zeros and ones have been generated as a code to be added to the images. The data has been encrypted an embedded in four text images as shown in figure 3. Three RGB images and one gray scale image has been selected. Figure 3 shows in the left the image before the data hiding process 'the cover image' and one the left the image after data has been embedded. Visually, the images look the same. However, to measure the changes that occurred visually on the images, peak signal to noise ratio (PSNR) has been utilized. To calculate the PSNR, mean square error (MSR) should be calculated. Eq.1 is used to calculate MSR and Eq.2 is utilized to calculate the PSNR.

$$MSR = \frac{1}{H * W}\sum_{i=0}^{H}\sum_{j=0}^{W}\left(X_{ij} - Y_{ij}\right)^2 \qquad (1)$$

$$PSNR = 10(\frac{255^2}{MSR}) \qquad (2)$$

Where 'Xij' and 'Yij' are the values of the pixels before and after data embedding process. 'H' and 'W' are the diminutions of the image. The MSR and PSNR of the proposed embedding algorithm have been compared to LSB [19] and turtle shell method [18]. Table 1, shows PSNR values and Tables 2 shows the MSR values.

**TABLE 1.** MSR VALUE

| Algorithm | 1st | 2nd | 3rd | 4th |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| PassDirectory | 54 | 50 | 42 | 32 |
| Turtle Shell | 361 | 668 | 142.1 | 486.9 |
| LSB | 0.23 | 0.23 | 0.25 | 0.26 |

**TABLE 2. PSNR Values**

| Algorithm | 1st | 2nd | 3rd | 4th |
|---|---|---|---|---|
| PassDirectory | 47 | 45.2 | 43.3 | 44.1 |
| Turtle Shell | 22.5 | 19.8 | 26.6 | 21 |
| LSB | 54.3 | 54.4 | 54.1 | 54.2 |

I can observe from the tables that PassDirectory has PSNR higher than the ordinary turtle shell algorithm. However, the value is lower than the LSB. In addition, we can observe that the data embedding capacity of the algorithm is 0.33bpp which is lower than both of the algorithms. However, the proposed system attempts to embed credential values, which are very small. Moreover, it has been reported by Gigaom that on average the number of photos that smartphones' users have on their phones is 630 photos [20]. This number of photos will be more than enough to embed the secret usernames and passwords without any capacity issues. It worth mentioning that, the data is embedded in more than 3 images and photos to reduce the loss impact if the photos are deleted. These three or four photos and images are selected according to their last modification date. Two photos should be old and another two should be new. We did not study the photos life time in users' smartphones.

Another strong point of the proposed system that the old cover image can be retrieved and all the embedded data can be deleted easily unlike the other embedding techniques.

## CONCLUSTION

Passwords steal the most popular authentication technique all over the Internet. Web-services, online banking, Emails and even gaming web-sites depend on usernames and passwords to identify and authenticate customers and users. Tens of sites and services and even hundreds are utilized and used by users in daily bases. This requires the users to generate new credentials for all of these sites and services or to reuse their old credentials used in other sites. To tackle the password memorization issue, password management systems are used. In this work, a new password management system is proposed for smartphones' platforms. The system hides usernames and password encrypted in images and photos on the smartphone. A new modified version of PlayFair symmetric encryption has been used. Subsequently, the images pixels are swapped to hide the encrypted data. We have compared the proposed system to LSB and turtle shell. Our result shows that the system has a high PSNR compared to the other systems. In the future, we attempt to install the system and tested in android phones to measure its power usage, storage capacity and delay of data embedding and retrieving. In addition, creating a new method to recovery from data losses that occurs when images are deleted.

## REFERENCES

1. M. Masoud, Y. Jaradat, . Manasrah, and I. Jannoud. Sensors of Smart Devices in the Internet of Everything (IoE) Era: Big Opportunities and Massive Doubts, *Journal of Sensors 2019,* 2019.
2. Avergae number of Apps in users' smartphone. [Online]. Available at: http://www.statista.com/statistics/267309/number-of-apps-on-mobile-phones/#:~:text=On%20average%2C%20U.S.%20users%20had%2020%20apps%20 installed%20on%20their%20mobile.
3. No wonder hackers have it easy: Most of us now have 26 different online accounts - but only five passwords. [Online]. Available at: http://www.dailymail.co.uk/sciencetech/article-2174274/No-wonderhackers-easy-Most-26-different-online-accounts--passwords.html.
4. Online Passwords: Keep it Complicated. http://www.guardian.co.uk/ technology/2012/oct/05/online-security-passwords-tricks-hacking.
5. D. Florencio and C. Herley. A Large-Scale Study of Web Password ˆ Habits, *in Proceedings of the 16th International Conference on the World Wide Web,* 2007, pp. 657–666.
6. A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, The tangled web of password reuse, *in Network and Distributed System Security Symposium, NDSS*, Feb 2014.

7. Millions of LinkedIn passwords reportedly leaked online. [Online]. Available at: http://news.cnet.com/8301-1009 3-57448079-83/millions-of-linkedinpasswords-reportedly-leaked-online/.
8. 6.46 million LinkedIn passwords leaked online. http://www.zdnet.com/ blog/btl/6-46-million-linkedin-passwords-leaked-online/79290
9. LinkedIn confirms password leak, encourages users to update passwords. [Online]. Available at: http://www.cbsnews.com/8301-501465 162-57448443- 501465/linkedin-confirms-password-leak-encourages-users-to-updatepasswords/.
10. Pwned websites, Have I Been Pwned, 2019. [Online]. Available at: https://haveibeenpwned.com/PwnedWebsites
11. M. Golla, M. Wei, J. Hainline, L. Filipe, M. Dürmuth, E. Redmiles, and B. Ur. What was that site doing with my facebook password?: Designing password-reuse notifications, *in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS*, 2018
12. A. Smith. Americans and cybersecurity. Pew Research Center. [Online]. Available at: http://www.pewinternet.org/2017/01/26/ 2-password-management-and-mobile-security, January 2017.
13. E. Stobert and R. Biddle. A password manager that doesn't remember passwords, *In Proceedings of the 2014 New Security Paradigms Workshop (NSPW),* 2014.
14. Ion, R. Reeder, and S. Consolvo. No One Can Hack My Mind: Comparing expert and non-expert security practices, *In Proceedings of the 11th Symposium on Usable Privacy and Security (SOUPS'15),* July 2015.
15. KeePass Password Safe. [Online]. Available at: https://keepass.info/.
16. Dashlan. [Online]. Available at: https://www.dashlane.com/.
17. 1Password. [Online]. Available at: https://1password.com.
18. C.Chang, Y. Chou, and The D, Kieu. An information hiding scheme using Sudoku, *In 2008 3rd international conference on innovative computing information and control, IEEE,* 2008, pp. 17-17.
19. K. Al-Afandy, O.S. Faragallah, A. ELmhalawy, E.M. El-Rabaie, and Gh. M. El-Banby. High security data hiding using image cropping and LSB least significant bit steganography, *In 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), IEEE,* 2016, pp. 400-404.
20. Average number of Images, [Online]. Available at: http:// power959.com/how-many-pics-does-the-average-person-have-on-their-phone/#:~:text=According%20to%20Gigaom%2C%20the%20average,photos%20stored%20on%20their%20phone.
21. K. Yee, and K. Sitaker. Passpet: convenient password management and phishing protection, *In Proceedings of the second symposium on Usable privacy and security,* pp. 32-43. 2006.