# GRABLOK: A Novel Graphical Password Authentication Utilising Blockchain Technology

Stavros Shiaeles
Cyber Security Research Group
University of Portsmouth, PO1 2UP
Portsmouth, U.K.
stavros.shiaeles@port.ac.uk,
https://orcid.org/0000-0003-3866-0672

***Abstract*— One of the most important security issues is unauthorised access to computer systems. The number of leaked passwords and credentials grows exponentially each year, showing that current protection systems and authentication methods are insufficient. Attackers are bypassing the state-of-the-art systems and gaining access to corporate environments as well as our personal accounts comprising confidentiality and threatening privacy. This work focuses on a new password authentication scheme utilising 3D graphical passwords and Hyper Ledger Fabric. The initial implementation shows that this method is promising and can offer users and organisations better security minimising the risk of stolen credentials.**

**Keywords—Graphical passwords, Privacy, Blockchain, Hyperledger Fabric, Authentication, Textual passwords,**

## I. Introduction

Despite a large number of options for authentication, text passwords remain the most common choice for many reasons [1]. The reasons vary, the most prominent being the simplicity, given that most users are familiar with the process and the inexpensive nature of implementing this authentication method. Unfortunately, passwords are prone to attacks such as phishing, dictionary attacks, credential stuffing, keyloggers and brute force attacks. Regarding brute-forcing attacks, someone could say it is due to the human nature of long-term memory (LTM) limitation, which leads users to choose simple or short passwords that are easy to remember. Short or simple passwords, along with the advancements in technology and currently available processing power, are helping attackers have easier and faster access to plain text passwords, which would significantly degrade authentication security. According to the 2020 Verizon data breach investigations report, 81% of the total number of breaches leveraged stolen or weak passwords [2]. However, with over 93% of authentication methods relying on passwords, passwords are clearly the most dominant form of authentication [3].

Even if most users have been educated or forced to follow basic rules in passwords creation, such as a minimum password length of 6 to 8 characters[4], using a combination of upper and lower case letters, using special characters, attackers find various ways to steal their credentials by lure them to fake websites or using different other methods such as man-in-the-middle. The internet is full of compromised password databases [4], and analysis showed that passwords that users create often contain specific phrases or words, including names, locations, dates and years [5], birthdays, addresses, pet names or children names. A hacker can easily access all of this information through social media and create wordlists for the target persons that can be used to crack their passwords[6].

There are various alternative methods to text passwords, with the most competitive, I would say the biometrics. Biometrics are a good alternative as it is something you already own, i.e. iris, hand vein, fingerprints, voice, face etc, and you do not need to carry anything like RFID cards. However, there are many privacy issues behind this, and many people are reluctant to provide these unique identifiers to their employers or online websites. It should be noted that if these biometrics are for any reason leaked, you cannot replace your iris or face as you can easily do with a text password. In this respect, new methods are needed for password authentication to respect users' privacy and also provide users with higher security confidence.

A very interesting approach and well-studied in literature are Graphical Passwords(G.P.). G.P. bypasses the LTM human limitation as it has been identified that humans generally have better memory and recognition capabilities for images than textual strings [7], [8]. This paper focuses on a new graphical password authentication method using Blockchain technology, specifically Hyper Ledger Fabric. The proposed method is described in detail as well as its advantages in terms of flexibility and security compared with alphanumeric passwords. Also, the proposed method's empirical evaluation of user performance and perceptions are provided.

The rest of this paper is structured as follows: Section 2 gives an overview of related work. Section 3 presents the proposed methodology. Section 4 discusses the current implementation. Section 5 evaluates the method against some attacks theoretically, and Section 6 concludes and provides the future direction of this research.

## II. Related works

Until recently, society considered security as only a technical problem. However, it is now becoming widely recognised that security is fundamentally a human-computer interaction (HCI) problem. Most security mechanisms cannot be effective without taking into account the user[9]. One of the critical areas in security research and practice is authentication, the determination of whether a user should be allowed access to a given system or resource. Even though various methods have been proposed, including biometrics, smart cards etc., smart cards also need a PIN or password, while biometrics have raised various privacy concerns. This has led the research community to look for new innovations to improve passwords' strengths while being easy for the user to memorise. One such innovation is graphical passwords, i.e., those based on images rather than alphanumeric strings.

The first idea for graphical passwords was described by Blonder[10]. His approach was to let the user click, with

a mouse or stylus, on a few chosen regions in an image that appeared on the screen. If the correct areas were clicked, the user was authenticated, otherwise, the user was rejected. Graphical password systems are based either on recognition, cued or pure recall. The proposed method is based on recognition, so we will focus our literature review on this area.

GOTPass[11], is a hybrid graphical authentication that authenticates a user using a one-time numerical code. The one-time code is produced by clicking two images on a 2D grid; you need to enter that code in any application you want to enter. The method was tested using simulated Guessing, Interception, and Shoulder-surfing attacks showing a promising 98% attack proof. However, this method is vulnerable to shoulder-surfing or Interception.

PassPoints [12], extended Blonder's idea by overcoming some of its main limitations. It eliminates predefined boundaries, and it is possible to use any image, such as natural images, paintings, etc. Since each image contains hundreds to thousands of potential click points, PassPoints has an enormous theoretical password space. Users find it difficult to click points within boundaries, taking more time than an alphanumeric password. PassPoints is vulnerable to shoulder surfing attacks since attackers can observe the click points directly during authentication.

Cued Click-Points (CCP) is a variation of PassPoints [13] where users click on one point per picture for a series of photos. The image displayed is related to the coordinates the user clicks on the previous picture. Users will not see the right picture if they do not click on the correct spot. From Chiasson et al. [14] analysis, it was found that users preferred to choose click-points that were located within well-known hotspots; thus, it is prone to attacks. To address this issue, authors proposed Persuasive Cued Click-Points (PCCP) [14], which push users toward using more random passwords. As part of the password creation process, the images are slightly shaded except for a small square viewport area on the randomly positioned image. To avoid clicking outside this viewport, users must choose a click-point inside it. Users can repeatedly hit the "shuffle" button until a suitable site is identified to move the viewpoint randomly. The proposed method does not have any more shading or the viewport issue CCP had. Moreover, PCCP users are guided to choose more arbitrary click points. Based on tests, the PCCP seems to successfully address significant security issues linked to frequent hotspots and patterns, expanding the useful password space and retaining usability [15]. Additionally, PCCP lessens the consequences of hotspots. A drawback for both CCP and PCCP is that users' passwords can still be cracked after the attacker captures the login process or input sequence [16], which does not solve the issue of shoulder surfing attacks.

Background Pass-Go (BPG) [17] enhanced the Pass-Go with background graphics to help users remember their passwords and lower the success rate of password guessing. Based on a combination of Multi-Grid DAS, Pass-Go, and BPG ideas we have a new method named Multi-Grid Background Pass-Go (MGBPG) [18]. By selecting a personalised background image and scaling the grid lines, MGBPG users can decrease memorability. To achieve a memorable password while maintaining high security, MGBPG must find a balance.

CBFG [16] uses the basic principles of PassPoints and the idea of picture identification. The system shows four background graphics and ten icons during registration, where the users need to choose at least one cell and one icon from each image to use as pass-cells and pass-icons, respectively. The login screen contains four background images with random numbers (0-9), an icon, and ten numeric buttons corresponding to 0-9. To log in, users should click any number button until the icon is the pass-icon, without the need for precise pass-cell order. Users should keep clicking the remaining numeric buttons after authenticating the pass-cells to ensure they have all been clicked and until the system outputs a successful or unsuccessful message. CBFG offers a substantial amount of password space with numerous background pictures. Even if an attacker records the entire login process with a camera, it is still difficult to guess the user's password because the sequence entered each time has strong randomisation, and the start time and end time are well hidden. The results of the experiments indicated that CBFG is highly capable of resisting shoulder surfing attacks and intersection analysis attacks. Nevertheless, some risks result from user behaviour, such as the time interval between button clicks.

Dhamija et al. [19] proposed Déjà vu in 2000. Users in this method, during registration, are prompted to select a certain number of random art pictures from a set of images generated by a program. When a user tries to log in, a grid of decoy pictures along with password pictures are shown, and the user must identify them. The disadvantages of Déjà vu include the difficulty of remembering an obscure picture and the smaller password space relative to alphanumeric passwords. In PassFaces [20] users need to click on face images pre-selected in registration for several such rounds. Relative literature reported serious security problems in PassFaces [21]. The presence of clear photos of faces renders the system vulnerable to shoulder surfing and spyware.

This paper proposes a scheme that mitigates all of the weaknesses mentioned above and offers improved security, usability, and memorability. This has been achieved by utilising a range of mechanisms and state-of-the-art blockchain solutions, in a novel manner which is explained below.

## III. Proposed method

This section describes the proposed method for GRABLOK. As shown in Fig. 1, the core aspect of the proposed system is the blockchain network that allows users access to various services.

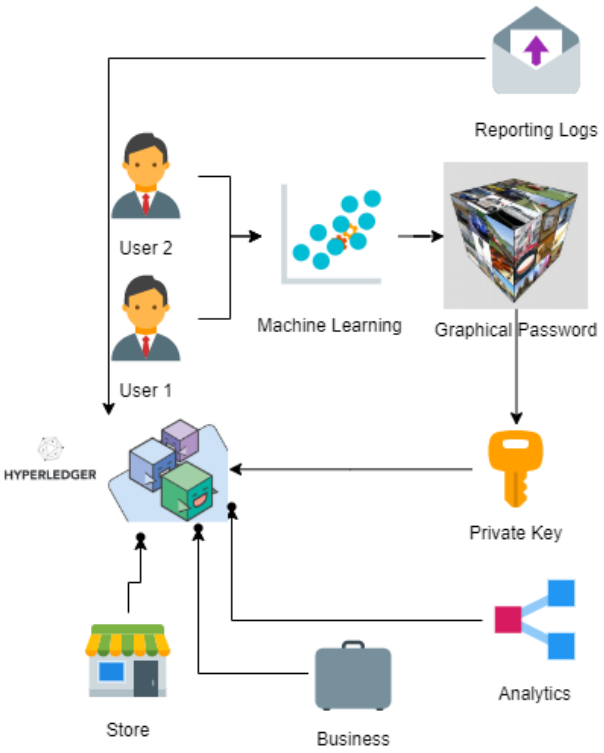

Fig 1. High-level architecture of the GRABLOK

There are other two important components for this solution. The first one is the machine learning, which is responsible for retrieving images from the web similar to the users selection or users uploaded images. The second one is the 3D cube which is responsible for authenticate users and add the information on the blockchain so that legitimate users access appropriate services.

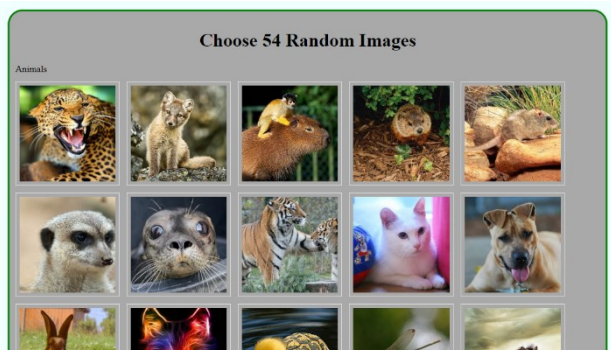

Fig 2. System proposed Images

In order the user to register, will need to provide an email or a unique username. Then the user is provided with the option to either select six(6) or more pictures from a pre-existing database or upload his/her own pictures of interest. The machine learning engine is reading the images and is retrieving similar, free images from the web in order to construct a database of fifty-four(54) total images, including the 6 of the user. The next step is user to choose a master image from a random database of images that is presented. This image will be used once and will produce the unique key for the device which will be named $K_{device}$. For each picture downloaded a random 12 character password is produced with upper, lower case, special symbols and numbers and is embedded in each picture using discrete wavelet transform (DWT) steganography and the $K_{device}$ as index for random storing the password characters inside the picture. Moreover, pressure and velocity are handled as 'hidden' features to increase the resiliency of the scheme. Once this step is completed, the 3D cube is constructed and can be used for user authentication.

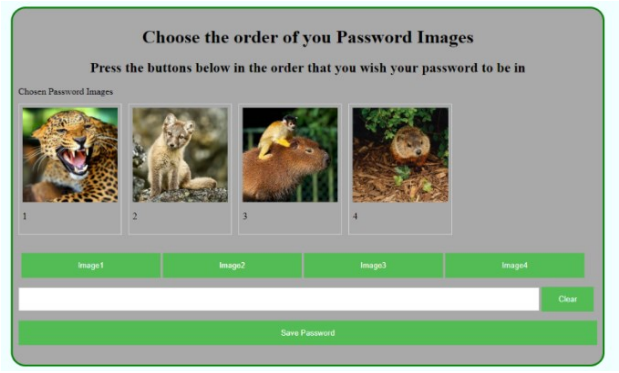

Fig 3. Images Click Sequence define

The user who has an account and wants to log in to the system is prompted to enter his/her email or username. Then the 3D cube retrieves the images from the local storage and waits for the user to click on the six(or more, depending on the user's preferences) pictures chosen with the proper order out of the fifty-four images shown on the 3D Cube sides. The images are randomly placed on the six sides of the 3D-Cube, and each time the cube reloads, they are reordered. Once the user successfully clicks the pictures, a temporary public key is produced using elliptic curve encryption, and it is stored on the blockchain along with some other encrypted information like username, login coordinates and expiration time in epoch. The public key and the information above will produce a unique QRCode that will be available for the user through the blockchain. Users can access the systems in the blockchain network until the public key is expired by using their private key. Then re-authentication will be needed. The blockchain network would not allow a second temporary key to be stored on the network if the one is not expired, making the password reuse difficult for attackers. This proposed method will be further validated against some attacks in section 5. Moreover, failed attempts and misuse of authentication will be reported and stored on the blockchain network.

## IV. Implementation

This section describes the current progress on the proposed solution. The project's first step was the development of the 3D interactive cube that would be used for graphical password authentication. The cube is an authentication method that allows users to select a sequence of images they previously chose as a text password alternative. A minimum of 6 images are uploaded or selected from the proposed system (Fig. 2) by the user and defines the sequence in which these images should be clicked (Fig. 3).

This procedure utilises reverse image API integration with Google Images and OpenCV Python Libraries able to recognise what is showing on the picture and find more similar images, confusing attackers. The cube has six sides, and the grid on each side has 9 (3 x 3) images. That means a total of 54 images are used for the 3D cube to be complete. Of course, the cube can be extended to anything beyond

3x3, which is up to the implementor. Our initial implementation was on 3x3.

The library used for creating the 3D cube is called Three.JS, which is a JavaScript library that allows 3D computer graphics to be made, displayed and interacted within a web browser. This library's documentation is accessible but does not have much detail on how each function is used and does not give specific examples of how to use the feature. On the Three.js website, however, there are many demonstrations of what the library is capable of, and that was our resource to make the 3D Cube (Fig. 4) applicable.

For the Blockchain network, Hyperledger Fabric (HLF) was chosen. Hyperledger, an open-source umbrella project of Linux Foundation has several significant projects that run simultaneously, enabling developers and businesses to experiment with blockchain networks at ease [23]. Hyperledger Fabric under the Hyperledger project provides features that perfectly fit this paper's use-case scenario. It is a permissioned and closed blockchain suitable for a consortium of services that allow users access; this solves the confidentiality issue of identity data to external parties. Fabric presents the concept of ordering service, Certificate Authority (C.A.) and Membership Service

Fig 4. 3D Cube Password Authentication in action

Providers (MSP), are used to identify and verify when a transaction request is made [22], [23]. Like Ethereum, Fabric provides the concept of smart contracts, which initialises and manages ledger state through transactions submitted by applications called Chaincode. Chaincode is business logic and is deployed on every endorsing peer [24]. The channel concept enables Fabric organisations to operate confidentially with the channel ledger containing data transactions only visible to channel members. Hyperledger Fabric implements an execute-order-validate mechanism i.e. the transactions are executed before reaching a final agreement on their order; then all peers validate the transactions in the same order with a deterministic validation [25]. Thus, Fabric in comparison to other blockchains, provides flexibility, scalability, and privacy. Although complex to implement and follow, Hyperledger Fabric has deep documentation with theory and tutorials, with the community growing in recent years. As a result, after careful consideration of the limitations of blockchain technology such as scalability, ledger size, performance and, a trade-off between decentralisation and identity data flexibility, Hyperledger Fabric permissioned blockchain was implemented in this project that enabled to maintain balance with the technology's limitations whilst development of a system prototype mentioned in this paper. Table 1 below shows a quick comparison between HLF and Ethereum. The Fabric test network and binaries can be downloaded from the reference [28].

| Characteristics | Ethereum | HLF |
|---|---|---|
| Description | Generic blockchain platform | Modular blockchain platform |
| Mode of Operation | Permissionless | Permissioned |
| Scalability | High node-scalability, low performance-scalability | Low node-scalability, high performance-scalability |
| Consensus | Mining based on PoW | Pluggable PBFT |
| Transaction per second (tps) | ~ 20 | >2000 |
| Smart Contract | Yes | Yes (Chaincode) |
| Language | Go, C++, Python | Java, Go |
| Currency | Ether | None |

Table 1: Comparison of HLF Vs Ethereum [20], [21]

Once a user is registered and private keys are produced, every time is authenticated a QR-Code is produced with a fixed expiration of 1 day using the private key along with the mouse pressure and velocity and is stored in the blockchain as transaction. This transaction is recalled when the user tries to access a resource and enters his/her email/username. Then using the phone's camera, the QR-Code is scanned and decrypted using the device's private key, providing the user access to the resources requested. It is also worth noting that the authentication is logged in the blockchain as well as the GPS position and I.P. for accountability purposes. If the public key is expired, the 3D-Cube will pop up, producing a new QR-Code after the authentication.

## V. Security Analysis of the Proposed Solution

This section will compare the proposed solution with some well-known attacks to show its superiority. The analysis is done theoretically as the entire method is not yet implemented and tested with real users.

### *A. BRUTE-FORCE ATTACK*

A brute-force attack is a method of finding a password by trying a large number of combinations. The time taken to break a password depends upon both the password length as well as the password space. The password entropy (E),

meaning how unpredictable a password is, is calculate from the following formula:

$$E = \log2(R)*L = \log2(R^L)$$

Where:

R - Size of the pool of unique characters from which we build the password; and L - Password length, i.e., the number of characters in the password.

In our case, R is equal to all characters of the English alphabet, both upper and lower case, as well as numbers and special characters. This make our R equal to 26 + 26 + 10 + 32 = 94. Our Password Length is equal to 12 x number of pictures selected, which is 6 in our case so the length is 72 characters long. The Entropy of our proposed system password is 471.93 bits. Fig. 5 below provides a nice view of password cracking based on brute force guesses per second and password entropy.

**Time until *guaranteed* brute-force password crack**

Based on attacker's guesses per second vs. password strength

**Formula**: (*Seconds to guaranteed crack*) = ($2^{(entropy)}$) ÷ (guesses per second)

Result then converted from *seconds* to more reasonable units of time such as years

**Note**: By definition, it takes **half** of the *guaranteed* crack time on *average* to crack a password

**Entropy** (in bits)

Entropy = $\log_2(S^L)$

**L** is pass length

**S** is symbol pool

*(i.e. if your password has lowercase and numbers, your **symbol pool** is 26+10=36)*

*(If entropy confuses you, just focus on the rest of the chart)*

| Attacker's brute force guesses per second | | | | | |
|---|---|---|---|---|---|
| 1,000,000 | 1,000,000,000 | 100,000,000,000 | 1,000,000,000,000 | 100,000,000,000,000 | |
| ∞ | ∞ | 421 quadrillion years | 42 quadrillion years | 421 trillion years | 120 |
| ∞ | ∞ | 105 quadrillion years | 11 quadrillion years | 105 trillion years | 118 |
| ∞ | ∞ | 26 quadrillion years | 2.6 quadrillion years | 26 trillion years | 116 |
| ∞ | 658 quadrillion years | 6.6 quadrillion years | 658 trillion years | 6.6 trillion years | 114 |
| ∞ | 165 quadrillion years | 1.6 quadrillion years | 165 trillion years | 1.6 trillion years | 112 |
| ∞ | 41 quadrillion years | 411 trillion years | 41 trillion years | 411 billion years | 110 |
| ∞ | 10 quadrillion years | 103 trillion years | 10.3 trillion years | 103 billion years | 108 |
| ∞ | 2.6 quadrillion years | 26 trillion years | 2.6 trillion years | 26 billion years | 106 |
| 643 quadrillion years | 643 trillion years | 6.4 trillion years | 643 billion years | 6.4 billion years | 104 |
| 161 quadrillion years | 161 trillion years | 1.6 trillion years | 161 billion years | 1.6 billion years | 102 |
| 40 quadrillion years | 40 trillion years | 402 billion years | 40 billion years | 402 million years | 100 |

Fig 5: Password Cracking Calculation based on hardware [29]

As seen from table 2, for 100 trillion guess per second, a password with 94 symbols (upper, lower case, numbers and special characters) and an Entropy of 118 needs 105 billion years to be cracked. In our case is 471.93 entropy which depending on the formula

$$\text{Seconds for Cracking} = 2^{(\text{Entropy})}/\text{Guesses per second}$$

will need a substantial amount of time to be cracked, assuming the password is not changed in a year or sooner. We also need to note that the password is entirely random, which is very important for the Entropy. Moreover, this attack can be done only offline as systems block multiple attempts, so the attacker will need access to the local private key, which demands a lot of effort and will be detected once used.

### *B. DICTIONARY ATTACK*

As the dictionary attacks name suggests, a password is cracked by comparing it with a pre-generated list of passwords found in the dictionary. As the password generated is entirely random and directly from the computer without user interaction, this attack would not be possible to succeed.

### *C. SHOULDER SURFING ATTACK*

A shoulder surfing attack consists of observing and recording the login activities of a user. Camera recording can be utilised to capture the password when the user types the credentials at the login screen. Due to the nature of 3D cube and that it takes into account the pressure and velocity as 'hidden' features to increase the resiliency of the system, this method will not work from the attacker side, especially from a remote authentication login as there is also delay in the 3D spin. Moreover, as the successful login session is stored on the blockchain and only one session is allowed per day, along with the geolocation of the valid user, the system will not allow a second login. Also, the attackers' attempt and details will be logged and stored on the blockchain. Finally, I consider the recording of the 3D cube extremely difficult.

### *D. RANDOM GUESSING ATTACK*

Because the chance of correctly guessing all password elements is meagre due to password randomness, 3D Cube login is not susceptible to random guessing attacks. Moreover, the password size is enormous making it impossible to guess.

### *E. PHISHING OR FORMING ATTACK*

Phishing attacks redirect users to a fake website and ask them for their passwords, where the attacker records their passwords. As the proposed system works only with validated websites that are registered on the blockchain, the proposed method is not vulnerable to this attack. Suppose an attacker adds a malicious link under a trusted website that already has access to the blockchain network. In that case, the attacker will only be able to get the transaction id from the blockchain, which allows the user access to the resource and not the password. Moreover, the attacker will not be able to have access anyway with this id as the private key is stored on the user's device.

### *F. KEYSTROKE/MOUSE LOGGER ATTACK*

A keystroke logger sends information about the keys pressed to an attacker, while a mouse logger sends the (x, y) coordinates of the mouse click positions. As no keystroke is used and due to the fact that random numbers are entered into the password field as well as pictures are randomly change in 3D space, secure login is resilient to keystroke/mouse logger attacks.

### *G. MULTIPLE RECORDING ATTACK*

This attack aims to crack a password by obtaining information from multiple login sessions by using spyware applications such as screen scrapers, keystroke loggers, or mouse loggers. It may be sufficient to record one login session for the easy login method to crack a password, but for the secure login method, it is necessary to be in the same place the current session takes place. If the session is not expired, an attacker cannot have access even if the correct pictures have been recorded. Moreover, as everything logged on the blockchain the administrator and the user will be notified in case of unauthorised access and the attempt will be blocked immediately.

## VI. CONCLUSION

In this paper, we have proposed and implemented a 3D graphical password authentication mechanism utilising HLF. The proposed system employs two smart contracts to

manage the identity and validity of users accessing services. The HLF channel feature ensures only valid members have access to the shared ledger as an added security control measure. The deployment of MSP brings in further control measures to verify new blockchain applicants. The ECDSA key PKI architecture provides a more diminutive size key and is easier to compute for compact IoT devices. The primary device uses the digital certificate generated by the Fabric CA server to authenticate.

The proposed solution can withstand various attacks due to the strong password generation and the session key in the format of QR-Code creation that is stored on the blockchain and the cryptography used. Only the 3D Cube part is currently developed, and the HLF network is set up. Further work is needed in order to connect the session key with the HLF and users to evaluate our proposed method which is part of our future work. HLF is capable, as discussed in section IV, of handling many transactions per second and would be ideal for busy systems and big companies. This initial solution can have many future extensions. For example, Homomorphic Encryption could be utilised, and pictures could be stored on the blockchain and compared with the images clicked by the user. Moreover, future work will test the proposed solution on a big scale and measure the authentication time needed. Of course, all of these suggestions need further implementation and tests that could be explored in a new paper.

## REFERENCES


[1] HERLEY,C.,VAN OORSCHOT,P.,AND PATRICK, A. 2009. Passwords: If we're so smart, why are we still using them? In Financial Cryptography and Data Security. Lecture Notes in Computer Science, vol. 5628, Springer-Verlag, Berlin.

[2] "2021 world password day: How many will be stolen this year?" https://www.secplicity.org/. [Online]. Available: https://www.secplicity.org/2021/05/04/2021-world-password-dayhow-many-will-be-stolen-this-year/

[3] S. Furnell and R. Esmael, "Evaluating the effect of guidance and feedback upon password compliance," Computer Fraud & Security, vol. 2017, no. 1, pp. 5–10, 2017.

[4] Saini, J.R., 2014. Analysis of minimum and maximum character bounds of password lengths of globally ranked websites. International Journal of Advanced Networking Applications.

[5] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay, "Measuring real-world accuracies and biases in modeling password guessability," in 24th {USENIX} Security Symposium ({USENIX} Security 15), 2015, pp. 463–481.

[6] J. True, "5 shocking insights into the social network habits of security professionals," thycotic.com. [Online]. Available: https://thycotic.com/company/blog/2017/05/30/5-shocking-insights-intothe-social-network-habits-of-security-professionals-and-infographic/

[7] Nelson, D.L., Reed, V.S. and Walling, J.R., 1976. Pictorial superiority effect. Journal of experimental psychology: Human learning and memory, 2(5), p.523.

[8] Shepard, R.N., 1967. Recognition memory for words, sentences, and pictures. Journal of verbal Learning and verbal Behavior, 6(1), pp.156-163.

[9] Wiedenbeck, S., Waters, J., Birget, J.C., Brodskiy, A. and Memon, N., 2005. PassPoints: Design and longitudinal evaluation of a graphical password system. International journal of human-computer studies, 63(1-2), pp.102-127.

[10] Blonder, G., 1996. Graphical Passwords. United States Paten 5559961, Lucent Technologies. Inc., Murray Hill.

[11] Alsaiari, H., Papadaki, M., Dowland, P. and Furnell, S., 2015. Secure graphical one time password (gotpass): an empirical study. Information Security Journal: A Global Perspective, 24(4-6), pp.207-220.

[12] S. Wiedenbeck, J. Waters, J. Birget, A. Brodskiy, and N. Memon. "PassPoints: Design and longitudinal evaluation of a graphical password system". International Journal of Human-Computer Studies, 63 (1-2): 102-127, 2005.

[13] S. Chiasson, P.C. van Oorschot, and R. Biddle. "Graphical password authentication using Cued Click Points". In European Symposium On Research In Computer Security (ESORICS), LNCS 4734, September 2007, pp. 359-374.

[14] S. Chiasson, A. Forget, R. Biddle, and P. C. van Oorschot. "Influencing users towards better passwords: Persuasive Cued Click-Points". In Human Computer Interaction (HCI), The British Computer Society, September 2008.

[15] Chiasson, S., Stobert, E., Forget, A., Biddle, R., van Oorschot, P.C.: "Persuasive Cued Click-Points: Design, implementation, and evaluation of a knowledge-based authentication mechanism". IEEE Transactions on Dependable and Secure Computing (TDSC),2012.

[16] X.Y. Liu., J.H. Qiu., L.C. Ma., H.C. Gao., etc., "A Novel Cued-recall Graphical Password Scheme", In sixth International Conference on Image and Graphics (ICIG), pp.949-956, 2011.

[17] L. Y. Por, X. T. Lim, M. T. Su, and F. Kianoush. "The design and implementation of background Pass-Go scheme towards security threats". WSEAS Transactions on Information Science and Applications, 5(6):943-952, June 2008.

[18] L. Y. Por and X. T. Lim, "Multi-Grid background Pass-Go". WSEAS Transactions on Information Science and Applications, Issue 7, Volume 5, July 2008.

[19] Dhamija R. and Perrig A., "Déjà vu: A User Study Using Images for Authentication", in Proceedings of 9th USENIX Security Symposium, 2000

[20] Sacha Brostoff, M. Angela Sasse, "Are Passfaces More Usable Than Passwords?:, A Field Trial Investigation, 2000

[21] D. Davis, F. Monrose, and M. K. Reiter, "On user choice in graphical password schemes", in Proceedings of the 13th Usenix Security Symposium. San Diego, CA, 2004.

[22] J. Sreenivas, K. Yelpale, and V. V. Kamath, "Blockchain Solution to Healthcare Record System using Hyperledger Fabric," 2021.

[23] H. Anwar, "Hyperledger: The Enterprise Blockchain," 2019. https://101blockchains.com/hyperledger-blockchain/ (accessed Jun. 28, 2020).

[24] Hyperledger, "Hyperledger Architecture, Volume 1," vol. 16, no. 4, pp. 4129–4136, 2017, doi: 10.3892/ol.2018.9166.

[25] A. Marcelletti and B. Re, "FabNet: an Automatic Hyperledger Fabric Network Wizard," Accessed: Nov. 16, 2021. [Online]. Available: https://hyperledger.github.io/composer/latest/.

[26] P. S. Sajana M; Sethumadhavan, M, "On Blockchain Applications: Hyperledger Fabric And Ethereum," International Journal of Pure and Applied Mathematics, vol. 118, no. 18, pp. 2965–2970, 2018.

[27] T. T. Kuo, H. Zavaleta Rojas, and L. Ohno-Machado, "Comparison of blockchain platforms: A systematic review and healthcare examples," Journal of the American Medical Informatics Association, vol. 26,

[28] "Prerequisites — hyperledger-fabricdocs master documentation." https://hyperledger-fabric.readthedocs.io/en/release-2.2/prereqs.html#prerequisites (accessed Dec. 31, 2021).

[29] Password Bruteforcing depending on password entropy https://www.reddit.com/r/dataisbeautiful/comments/322lbk/time_required_to_bruteforce_crack_a_password/. Last access 08/09/2022