ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940www.ijsem.org

Vol 11, Issuse1 Jan 2020

Animated Password Verification System that Uses Past Internet Activity

MANGALAPALLI MANISHA[1] , ATHUKURI DHARANI[2] , MANDALA CHANDANA[3]
KALLU MOUNIKA[4]
SUPERVISOR , Dr M MURUGESAN

Abstract

This article recommends PassPage, a graphical password authentication technique with two-factor authentication for improved website security. It makes advantage of the user's latent knowledge gleaned from their online activity logs. In addition to entering a text password, the server always sends back nine tiny pages as a challenge whenever a user attempts to log in. Twelve people participated in our user trials. The findings of the experiment demonstrated that when users are acquainted with the login procedure, the average login success rate on a news website is consistently above 80%, and this percentage does not significantly drop after 6 days.

1 Introduction

Most websites nowadays still rely on the tried-and-true methods of user authentication—a username and text password. Unfortunately, text passwords have both length and strength limitations that make them vulnerable to hacking attempts if only used. Employing methods such as guessing, shoulder surfing, dictionary cracking, and so on. As an additional layer of security, some websites may send a dynamic code to a user's phone or in an email. Although this form of two-factor authentication is the most secure, it places a heavy burden on the user by requiring them to remember multiple passwords, log in to multiple accounts, and wait for long periods of time between each step. An additional, more convenient authentication element is graphical password authentication. The majority of currently available graphical password authentication techniques, however, do not do away with the need for the user to establish and remember a password. If you utilize them across several websites, you'll need a lot more RAM to keep up. Our proposed authentication solution shouldn't add any more work for the user in terms of either operations or memory requirements. Our plan is to make advantage of the information the users already know about the site. It ought to stick in the user's mind organically when they peruse the web. Additionally, its entropy should be quite high. The pages the user visits on many websites, such as news sites, social networking sites, video sites, forums, and blogs, satisfy our requirements. In this setup, the website uses page scripts to automatically record the user's browsing history, which is then stored on the website's server for a predetermined amount of time.

Professor[1,2,3,4]
ANURAG ENGINEERING COLLEGE
AUTONOMOUS
(Affiliated to JNTU-Hyderabad,Aprroved by AICTE-New Delhi)
ANANTHAGIRI (V) (M), SURYAPETA (D), TELANGANA-508206

In order to log in, the user must choose from a list of 9 mini-pages that represent websites they've previously visited from the server's perspective. The user must also provide a text password, which acts as a second element of authentication. This kind of two-factor authentication is more secure than the standard method of authenticating using a text-based password, without significantly raising the computational or storage requirements. We've dubbed this method "PassPage" since the presented graphical passwords are really screenshots of websites. We built an experimental authentication system and tested PassPage with 12 volunteers to determine its usability. We built an extension for Chrome that secretly keeps track of users' surfing habits, and without touching the original websites' source codes, we created mock versions of signup, login, and password reset pages to ensure that all experimental data is delivered to our experimental server. The findings shown that when users are acquainted with the authentication method, the average login success rate on a news website remains consistently over 80% and shows no sudden decline in success rate over the course of 6 days.

2 Related Works

Our primary focus is on merging textual password authentication with graphical password authentication to create a more secure system. Many studies, from recall-based to cryptanalysis-based, have been conducted on the topic of graphical password authentication [1–13]. Mode, and the mode based on recognition. A number of scholars [2, 3, 6, 8, and 10] have developed alternative visual authentication methods. Some academics have been trying to make graphical authentication more secure [7, 8, 9, 12, 13], particularly against shoulder surfing attacks. Users must commit high-entropy secrets to memory to improve the security of graphical password authentication. However, we have no plans to increase our users' memorization requirements. This means that the suggested graphical authentication technique relies on the user's subconscious recollections. Some studies [14–18] have discussed implicit memory-based authentication. Implicit memory was initially suggested for authentication [14] by Tamara Denning et al. The author suggests the following criteria for a successful implicit memory authentication scheme: the secret can be stored in the brain for an extended period of time; the user is not required to remember anything on purpose during either the registration or authentication processes; the secret is random and has a high entropy; the processes of forming the memory and authenticating it are distinct. The authors provided a solution, however the experiment did not yield promising results; just 7% more accurate responses were received from participants who had seen all of the visuals. Users must also invest considerable effort into learning this system.

In-depth research and analysis of people's everyday memories was undertaken by Sauvik Das et al. [15]. The experimental users properly answered just 1381 (approximately 64%) of the 2,167 tasks (daily) originating from mobile phone data, the vast majority of which are recognition difficulties, and the correct rate is unaffected by time. The results of the poll suggest that users are more truthful about social concerns (such as the frequency with which they check their phones) than they are about their mobile phone use (in terms of both time and length). However, it is only useful when password authentication fails because the authentication process takes too long, on average more than a minute.

Based on consumers' familiarity with the programs pre-installed on their mobile devices, Huiping Sun et al.'s PassApp [16] created a mobile phone unlocking mechanism. The system takes 16 applications, 4 of which are already on the phone and 12 that aren't, and arranges their icons in a completely random sequence. To get access to the phone, the user must first press and hold each of four different app icons. When it comes to convenience and safety, this system excels. An impressive 95% of the time, our testers can recall 89% of the apps they've installed on their phones. However, you can only use this method to unlock a mobile phone. Pass Frame, introduced by Ngu Nguyen et al. [17], employs a tiny camera worn on the user's person to capture all external stimuli. Some still images from the video are extracted for user authentication when this is necessary. For its adaptability, it may be used in using what a user has seen as a kind of authentication is a viable option because of the authentication strategy based on implicit memory. However, it is impractical because it is so intrusive to people's privacy.

Simon Woo et al. advocated using a textual password based on life experiences (LEPs) [18]. Before logging in, users must select a security question and enter a textual response based on events in their lives such as birth, party, graduation, wedding, traveling, and others. Fuzzy matching is used by the system to verify the user's identity. A LEP is 30-47 bits more secure than a standard 8-character password. Almost no one can crack your password using brute force or a dictionary attack, and even your buddies have a 0.7% chance of figuring it out. None of these implicit memory systems make advantage of the user's contextual understanding of the page. PassPage, in contrast to these techniques, may be used for authentication on websites, and the secret pool grows as the user browses.

3 Design of PassPage

Sign-up, Record Browsing History, Decoy Web Page Maintenance, and Login are the four core system components that make up PassPage. In Fig. 1 we see their hierarchical relationship.

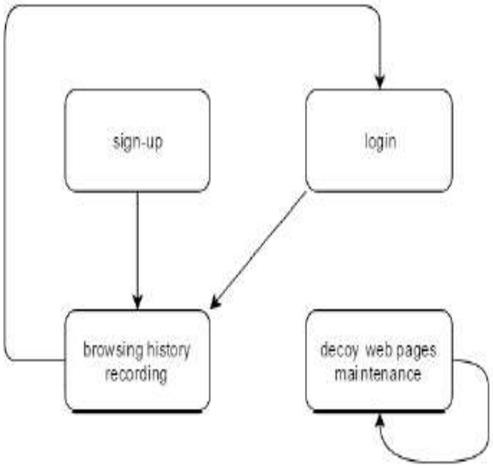


Fig. 1. Order relation of system modules

3.1 Sign-up Module

When creating an account, the user is asked for contact information (email address, username, and password). In case the user loses their passwords, they must provide an email address. A transmission to the server is made. The server enters the data into the database only after it has verified its veracity. Sends the client back into a session. The customer will then be logged in to the site mechanically.

3.2 Browsing History Recording Module

The module that records the user's browsing history activates as soon as the user enters into the website. When the page loads, a client script is executed in the background. The script first determines the user's identity by monitoring their session before saving the user's click stream. Some frequently visited but poorly recognized pages (like the website's homepage and information pages) are not tracked. If saving is required, the script will send the page's HTML code and the user's credentials to the server. The HTML content is saved on the server in a file with a random file name, and the username and file name are both recorded in the user page database. After then, the client receives the name of the file.

In addition, if the page needs to be recorded, the script will continue to record the time the user spends on the page even if they scroll up or down. The script will upload the file name and the timestamp array to the server when the user quits the page. The server records the user's total time on the site, the total number of pages scrolled, and a time array.

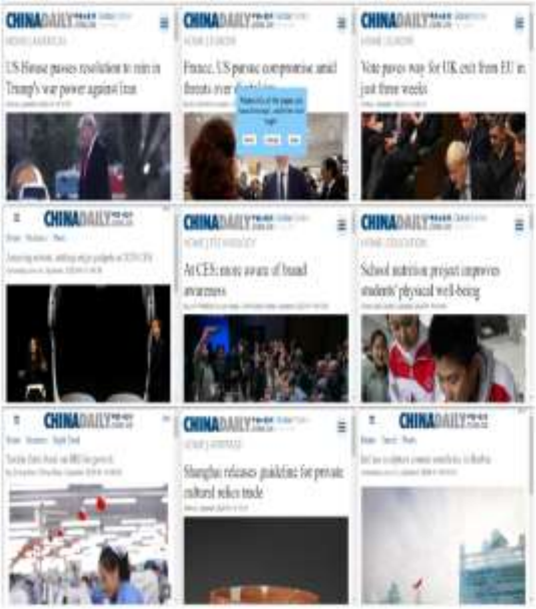
3.3 Decoy Web Pages Maintenance Module

When a user signs in, the server should provide them with a challenge that includes both legitimate and fake web sites. The server needs complete access to the webpage. Web sites must be requested once per every day and save them in the database of "decoy pages." It's preferable to have a wide variety of page themes; however pages with a high click rate but poor recognition rate shouldn't be kept.

Each page is saved to an HTML file on the server, and the server logs the page's name, title, and modification time. Decoy pages may be periodically erased to prevent an excessive growth in their total number. Assuming 500 new decoy pages are added daily, the server will search the database for pages that were added between 3 and 10 days ago and will delete half of them at random. As time goes on, the number of stale decoy pages will decrease but will never reach zero. There's a good chance that a few pages will stay put. In order for the long-term non-login user to view decoy web pages added long time ago, the server must return decoy sites with adding timings near to those of correct pages when the user tries to log in. In the following, we'll go through the method used to choose which pages to display.

3.4 Login Module

After entering their login and password, users may go to the next stage by clicking the "next step" button. Nine 3-by-3-inch miniature pages will be shown underneath the main layer. The bottom layer may be navigated vertically by the user. Login page implementation example shown in Fig. The brief documents might be altered ahead of time.



CHINADAILY

US House passes resolution to rein in Trump's war power against Iran

France, US pursue compromise amid threats over

Vote paves way for UK exit from EU in just three weeks

At CES: more aware of brand awareness

School nutrition project improves students' physical well-being

Shanghai releases guideline for private cultural relics trade

Figure 2: A sample login page (after the selection of "next step").

The user must click on each of the 9 pages and mark those that they have already seen. The user may then log in after choosing all of their previously viewed sites. Assuming visual verification is successful, passing the test and entering the proper password results in a successful login. If the login fails, the user may try again or choose other sites by clicking the "change" button. There are only three attempts per user to successfully log in. After three unsuccessful tries, the server will reject the login and prompt the user to reset their password using email authentication. When a user presses the "next step" button, the client communicates the user's credentials to the server. Passing the password test is necessary before moving on to the visual verification process. The user ID is sent to the server by the client. The client asks the server for nine HTML files by name, which the server then sends back to them (see Page Selection Algorithm). The user's browser will show 9 web sites, some of which will be familiar from previous visits (known as "real pages") and others that will be new (known as "decoy pages") to the user. When a user selects "login," the client communicates that information to the server along with the names of the chosen files. The server will return a session to the client if the authentication is successful.

An Algorithm for Choosing Pages. The server starts by gathering information about all the sites the user has viewed, including file names, titles, browsing logs, and the time they were added to the allRealPages collection. If size is 0, then the email authentication must be utilized instead of graphical authentication. If size is less than five, the server will choose two pages at random from allRealPages. If the size is more than six, then three pages from allRealPages will be used. The server has to choose legitimate sites that the user is more likely to know and remember. The server then pulls in decoy pages from the decoy page database, checking to see that the titles of the chosen decoy pages are distinct from allRealPages and that the adding times of the chosen decoy pages are near to the real pages'. The server deletes a total of 9 pages. Since each page is loaded from an HTML file, their combined size is less than 1 KB, and the loading time for all 9 pages is less than a second. Each user must check in before the server can allow them to change pages again, which prevents the attacker from cycling through pages to identify the consistent fake ones. This determines a limit on the user's total number of failed login attempts. If the user enters an incorrect password three times in a row, the account will be locked until a new visual password is created.

4 User Experiments

4.1 Experiment Procedure

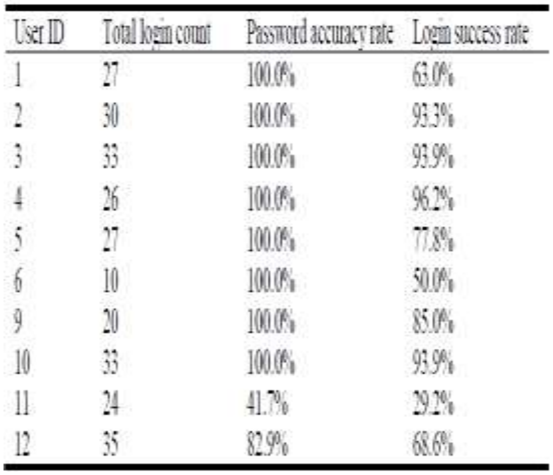
We built an experimental authentication mechanism to see how well PassPage would operate in real-world use. We built the server in Java and used HTML, CSS, and JS to create the sites and Chrome add-on. These pages are a practice sign-up form, login/password recovery page. Users' activity online may be tracked by installing an extension for Google Chrome. For this evaluation, we selected the news website www.sohu.com. Our experimental server receives all the data for the experiments, while the original website's code remains unchanged.

We advertised for new staff on several internet forums and message boards. No of your age, gender, field of study, or level of computer expertise, we welcome you to participate in our studies if you have 10 minutes to spare each day for the next 6 days. Any participant in the experiment is welcome to sign an agreement to participate. We eventually got 23 people to accept to participate. They range in age from 19 to 29. About half of them are studying to become software engineers or computer scientists.

We created a web page with instructions for the experiment in great detail so that participants could carry it out independently. On day one, they adopted our plugin for the Chrome browser. After that, they went to the mock registration page and set up fake accounts. Then, for a while, they just clicked around www.sohu.com. From day 2 through day 6, they repeatedly tried to log in using their test accounts on the dummy login pages. Every user was required to sign in five times daily. After that, they spent a few minutes looking around www.sohu.com. The system meticulously documented every user's login attempt. Each participant filled out a feedback table with seven remarks on their user experience when the trial was complete. There are five degrees of agreement after each statement. Each participant rated their degree of agreement with each statement on a scale from 0 to 10. Nine participants finished the experiment. They were all rewarded with fifty Chinese Yuan. Three participants abandoned their tests halfway through. They were each given ten Yuan (or Chinese dollars).

4.2 Experiment Results

In total, the volunteers tried logging in 277 times. We began by determining the login success and text password correctness percentages of participants who had more than 5 logins. Two-factor authentication relies on the precision of both the password and a physical token. Login is considered successful if and only if graphical authentication is successful. Rates of correct password entry and successful login attempts

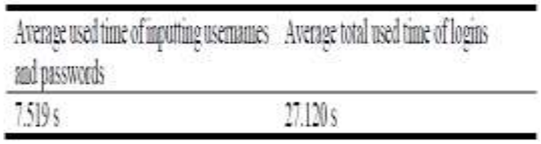


User ID	Total login count	Password accuracy rate	Login success rate
1	27	100.0%	63.0%
2	30	100.0%	93.3%
3	33	100.0%	93.9%
4	26	100.0%	96.2%
5	27	100.0%	77.8%
6	10	100.0%	50.0%
9	20	100.0%	85.0%
10	33	100.0%	93.9%
11	24	41.7%	29.2%
12	35	82.9%	68.6%

Success rates for logging in varied widely between volunteers, as seen in Table 1.

Four of the volunteers have a login success rate of at least 90%, while two have rates below 50%. We found that User 6 was unaware of the number because the actual number of pages was always 3, but he only used 1 or 2 of them, his login success rate is only 50%. User 11 was so negligent that he often tried to log in with the erroneous password since he had forgotten his text password. We also determined the typical duration of a login session, including the time spent entering a username and password.

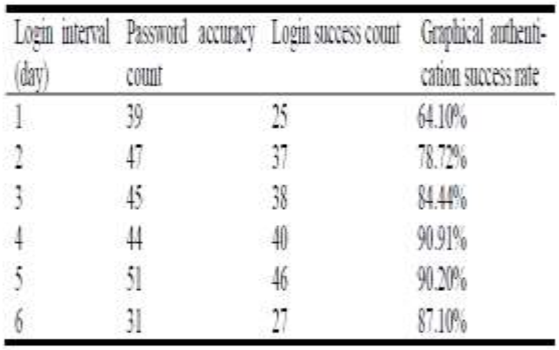
Table 2: A Look at How Long Each Authentication Method Is Typically Used



Average used time of inputting usernames and passwords	Average total used time of logins
7.519 s	27.120 s

Table 2 shows that the average amount of time spent logging in is around four times longer than the time spent typing in a username and password. Graphical authentication typically takes around 20 seconds. In contrast to competing visual authentication methods it’s a fair price, especially considering the use of dynamic codes for authentication. We then determined the percentage of graphical authentication that was successful as the time between registration and login became longer. Table 3 shows the results of our analysis of the login data, which we organized by the time that passed between registration and the first login (the "login interval").

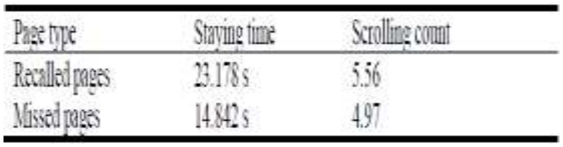
Table 3: Visual representation of login success rates by time since last login



Login interval (day)	Password accuracy count	Login success count	Graphical authentication success rate
1	39	25	64.10%
2	47	37	78.72%
3	45	38	84.44%
4	44	40	90.91%
5	51	46	90.20%
6	31	27	87.10%

Table 3 shows that the first-day success rate was much lower than expected. It could be due to the volunteers' lack of familiarity with the program. The success rate of graphical authentication was consistently above 80% between days 3 and 6. Inferences may be drawn about consumers' ability to recall recent online activity after just 6 days. We also determined how often graphical authentication was successful as the number of authenticated sites grew. Even with an increase from 35 to 40 actual pages, the success rate remains at 88.89%. We also determined how long people typically stayed on the site and how many times they scrolled over the page while thinking about remembered and forgotten actual sites. The duration a genuine page is open in the active window is known as its "staying time." The number of times an actual page has been scrolled is the sum of all of those times. In Table 4 you can see the final findings. We may infer that consumers are more likely to remember content from sites they spent more time on and scrolled through more times.

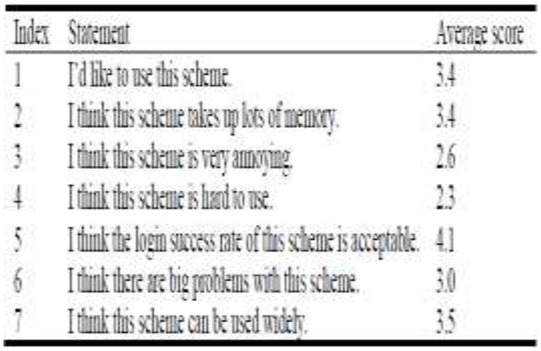
Average viewing duration and number of genuine pages remembered vs. missed



Page type	Staying time	Scrolling count
Recalled pages	23.178 s	5.56
Missed pages	14.842 s	4.97

The feedback tables' responses have been tallied. Table 5 displays the average rating for each assertion.

Table 5: Overall statement averages



Index	Statement	Average score
1	I'd like to use this scheme.	3.4
2	I think this scheme takes up lots of memory.	3.4
3	I think this scheme is very annoying.	2.6
4	I think this scheme is hard to use.	2.3
5	I think the login success rate of this scheme is acceptable.	4.1
6	I think there are big problems with this scheme.	3.0
7	I think this scheme can be used widely.	3.5

In general, the views of our volunteers are positive. The vast majority of test subjects agree that the scheme's login success rate is satisfactory, and that it is very simple to implement.

5. Widespread Use of Pass Page

For the PC browser version of the news website www.sohu.com, we created a special testing methodology. There is still considerable work to be done before it can be used extensively across many different websites and platforms.

5.1 Pass Page on Multiple Websites

Pass Page is flexible enough to be used on a number of different sites, but it requires customization for each one. Finding high-entropy information on each user on each website is of paramount importance. It might be the items the customer is interested in purchasing on a shopping website. Acquired or viewed in great detail. The user may have seen the uploaded pictures or videos on the social networking site. It might be the positions the user has applied for on a job board. A user's financing items purchased on a bank's website. Passwords in pages may be generated from these sorts of implicit memories. In reality, Pass Page is ineffective or unsuccessful across the board due to site restrictions. If the level of authentication difficulty is lowered, however, this authentication factor remains a preference on most websites.

5.2 Pass Pass on Multiple Platforms

Additionally, PassPage is cross-platform. Since the user's history is associated with their username and stored on the server, they need only enter their credentials once to access it from any device, be it a desktop browser, tablet app, or mobile phone. Developers of the property must provide provisions for their program or app that tracks users' activity on these services. They must also create platform-specific login screens.

6. Security and Privacy

Password authentication is made more secure by our technique. Let's worry just about how safe the graphical password is. Assume an attacker has learned a user's username and password and now wants to get access to the account via graphical authentication. The enemy is only privy to the true between 1 and 3 pages. The opponent has no further information to simply bypass the graphical authentication. He picks 1 out of 3 pages at random, giving him a total of C(9, 1) + C(9, 2) + C(9, 3) = 129 possible answers, of which only one is accurate. His chance of getting it right is thus just 1/129, or 0.775%. In order to lawfully gather surfing data, the website must first reach an agreement with users in advance, which might potentially compromise their privacy. It is important that these files be only accessible by the website server. In addition, the server database where users' browsing history is stored needs to be secured, or at least partially encrypted.

7 Conclusions and Future Work

In this work, we present PassPage, a two-factor graphical password authentication mechanism for online accounts. Users' subconscious recollections of previously visited websites are put to use. From what we could tell through our experiments, the typical rate of successful logins rate on a news website is consistently above 80% when users are acquainted with the login procedure, however an adversary with a random pick has just a 0.775% likelihood of passing the graphical authentication, not to mention the adversary needs to enter a correct password. Our tests also demonstrated that the login success rate did not drop significantly over the course of 6 days. We may draw the conclusion that these three goals are all met by this approach. To make the graphical authentication challenge more user-friendly, we want to do more research on how users interact with websites in the future. The server, for instance, should determine which sites the user is most likely to recall. Furthermore, there is still room for improvement in the login process to boost the login success rate.

References.

1. Biddle Robert, Sonia Chiasson, and Paul C. Van Oorschot. "Graphical passwords: Learning from the first twelve years." ACM Computing Surveys (CSUR) 44.4 (2012): 19.

2. Brostoff, Sacha, and M. Angela Sasse. "Are Passfaces more usable than passwords? A field trial investigation." People and computers XIV—usability or else!. Springer, London, 2000. 405-424.

3. Bianchi, Andrea, Ian Oakley, and Hyoungshick Kim. "PassBYOP: bring your own picture for securing graphical passwords." IEEE Transactions on Human-Machine Systems 46.3 (2015): 380-389.

4. Uellenbeck, Sebastian, et al. "Quantifying the security of graphical passwords: the case of android unlock patterns." Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013.

5. Stobert, Elizabeth, and Robert Biddle. "Memory retrieval and graphical passwords." Proceedings of the ninth symposium on usable privacy and security. ACM, 2013.

6. Zhu, Bin B., et al. "CAPTCHA as graphical passwords—a new security primitive based on hard AI problems." IEEE transactions on information forensics and security 9.6 (2014): 891- 904.

7. Gao, Haichang, et al. "A survey on the use of graphical passwords in security." JSW 8.7 (2013): 1678-1698.

8. Rao, Kameswara, and Sushma Yalamanchili. "Novel shoulder-surfing resistant authentication schemes using text-graphical passwords." International Journal of Information and Network Security 1.3 (2012): 163.

9. Renaud, Karen, et al. "Are graphical authentication mechanisms as strong as passwords?." 2013 Federated Conference on Computer Science and Information Systems. IEEE, 2013.

10. Khan, Mudassar Ali, et al. "g-RAT| A Novel Graphical Randomized Authentication Technique for Consumer Smart Devices." IEEE Transactions on Consumer Electronics 65.2 (2019): 215-223.

11. Mackie, Ian, and Merve Yıldırım. "A novel hybrid password authentication scheme based on text and image." IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Cham, 2018.

12. Mokal, P. H., and R. N. Devikar. "A survey on shoulder surfing resistant text based graphical password schemes." International Journal of Science and Research (IJSR) 3.4 (2014): 747- 750.