

NOW YOU SEE ME, NOW YOU CAN'T LOGIN

Natis Boon Jiabin¹, Joshua Ace Tong Zhi An², Sim Siang Meng³, Choo Jia Guang³

¹CHIJ St. Nicholas Girls' School, Ang Mo Kio Street 13, Singapore 569405

²Victoria School, 2 Siglap Link, Singapore 448880

³DSO National Laboratories, 12 Science Park Dr, Singapore 118225

Abstract

Shoulder-surfing is one of the simplest yet efficient methods to acquire sensitive information by spying on the victim's screen and input during an authentication process. Conventional password systems that use a fixed login passcode, e.g. a 6-digit PIN or pattern lock, will be trivially broken when the adversary observes the victim inputting the passcode even for once. Observer Resistant Password Systems (ORPSs) are authentication processes designed to prevent the adversary from learning the secret password even when they observe the victim's authentication process. In this paper, we look at some of the existing ORPSs, their pros and cons, and explore the development of a new user-friendly ORPS with quantifiable security.

1. Introduction

Security begins with authentication, which is determining whether or not to allow a certain user to gain access to potentially sensitive information or access to certain systems or resources. Shoulder surfing enables an attacker to obtain crucial information from the user, enabling them unauthorised access to important resources. This proves to be an important problem, since many existing ORPSs are susceptible to just 1 observation. Adequate authentication is always the first line of defence when protecting important resources. Currently, most conventional authentication processes are alphanumeric and carried out via users keying in their Username and Passwords directly into entry boxes. These conventional methods of authentication are susceptible to even just one observation.

Numerous ORPS have been proposed, each claiming to have high levels of quantifiable security, but they tend to not be usable for the average human in the general public. There are also cases where the opposite happens, where the proposed ORPS has reasonable usability but lacks sufficient security. Certain authentication systems require external hardware and are costly, such as biometric-based authentication systems. While the concept of other schemes such as graphical-based authentication systems may still be foreign to the average human of the general public, hence they feel unfamiliar using such schemes.

Seeing that the vast majority of the general public are most familiar with alphanumeric ORPS, in this paper we will propose an alphanumeric ORPS that is user-friendly with quantifiable security. The rest of the paper is organised as follows. In section 2, we will review related works. In section 3, we will discuss various ORPSs we had considered. In section 4, we will describe our proposed scheme. In section 5, we will analyse the security and usability of our proposed scheme. Conclusions and ideas for further works are made in section 6.

2. Related Works

As most people are familiar with conventional textual or alphanumeric password authentication schemes that do not have shoulder surfing resistance, Zhao et al. [1], in 2007, proposed a text-based shoulder surfing resistant graphical password scheme, S3PAS, in which the user has to find his or her textual password and then follow a special rule to mix his or her textual password to get a session password to successfully log in. However, the login process of the proposed scheme is complex and tedious. In 2011, Sreelatha et al. [2] also proposed a text-based shoulder surfing resistant graphical password scheme which involved the use of colours. Unfortunately, as the user has to additionally memorise the order of several colours, the memory burden of the user is high, causing usability of the scheme to decrease. In 2021, Jianwei Lai [3] introduced a password scheme similar to conventional alphanumeric authentication systems except while the user inputs his or her password, they are prompted by the system to omit certain characters in their password being entered into the system. This scheme, while being highly usable, its security needs to be further improved as the password of the user could be deciphered by cross-checking different recordings of the login process. In 2013, Mun-Kyu Lee [4] had proposed a scheme making use of multiple number pads. However, we found that the scheme was vulnerable to recording attacks, in which the authentication information is obtained by recording the authentication sequence for later analysis to open the device. In 2009, Gao et al. [5] proposed a shoulder surfing resistant graphical password scheme, ColorLogin, in which the background colour of the login interface of the scheme is also a factor for reducing the logintime. However, the probability of accidental login of ColorLogin is relatively high and the password space is relatively small. These reasons can cause the scheme to take a short time to be cracked. Though it proves to be fully resistant to shoulder surfing attempts, through multiple observations and eliminations, the password can be easily guessed. In 2014, Manjunath G [6] proposed an improved text-based shoulder surfing resistant graphical password scheme which also uses colours. Though the scheme had multiple factors to introduce uncertainty to confuse observers, in the worst case scenario, within 2 observations the password space of the scheme will fall to a worryingly small amount. The scheme could also possibly be susceptible to a dictionary attack. In 2011, M Sreelatha [7] proposed 2 schemes. For the first scheme, the user has to memorise 2 different passwords, which increases memory burden, affecting usability. For the second scheme, users have to rank and memorise the rankings of 8 colours, on top of their password. This also increases the memory burden of users.

From our analysis, we have found that the security of an authentication system can be increased vastly via three principles. Firstly, increasing the maximum key space of the system. Key space can be defined as the total number of possible values of keys in a cryptographic algorithm or other security measure such as a password. With an increase in key space, the success rate of a brute force attack decreases.

Secondly, minimising the decrease in key space per observation made by the adversary. This can be achieved by reducing the information that an adversary can gain from their observations. Under such a condition, even after a certain number of observations, the authentication system would still have a decent number of key space to still provide sufficient security.

Lastly, introducing uncertainty to increase ambiguity in the system. Inbuilt uncertainty is not within observer control, and will confuse observers, potentially decreasing the accuracy of the

conclusions they draw from their observations. The introduced uncertainty should have an increased complexity for an observer to solve for the password, such that it cannot be removed via statistical analysis based on probability difference.

Under such conditions, observers can only try to solve the password by removing the uncertainty via random guesses. The authentication scheme should then be designed to provide sufficient security even if the observer were to resort to such methods.

3. Schemes Considered

Before the finalisation of our scheme, we came up with multiple different password scheme ideas. We will analyse some of them in this section.

Colour Wheel Scheme

One of our ideas was the Colour Wheel Scheme (CWS), in which we implement the idea of getting users to input intentional wrong answers and equations with variables such as 'n' of which only the user knows the value of, to introduce uncertainty.

In the proposed scheme, during registration the user will rank multiple colours. Additionally, there will be a single wheel with 6 sectors of equal size distributed around the wheel. The user has to assign each colour to one sector. Users can choose to login with 6 or 12 colours. If the user decides to choose 12 colours, each of the 6 sectors in the wheel will be split into half, hence creating 12 positions equally distributed around the wheel for the user to assign their chosen colours to. Additionally, the user has to select a value for a variable 'n'. The value of 'n' ranges from 0 to 9 and it is only known by the user. The user has to memorise 5 variables, the chosen value of 'n', the number of selected colours to use during login, each colour, its rank and its position on the wheel.

In the login phase, the interface will have a single wheel with the colours selected by the user randomly distributed among the 6 sectors in the wheel. On the upper left corner of the interface, an equation involving the variable 'n' will be generated. The end value of the equations will represent the ranking of the colour to be verified. If the end value of the equation is less than or equal to the number of colours they had selected to use to login and greater than 0, the user will rotate the wheel to move the corresponding colour he or she had assigned the ranking to, into its correct position and click a 'confirm' button. However, if the end value of the equation is a decimal, or is less than 1 or is more than the number of selected colours to be used during login, then the user inputs an intentional wrong answer by moving the wheel randomly to jumble up the positions of the colours before clicking the 'confirm' button. This is one challenge, after 6 rounds of challenges, the user will login successfully. E.g. The user had selected 6 colours during registration and the generated equation gives an end value of 3, during registration, the user has registered blue as rank 3 and assigned blue to position 4 of the 6 positions (sectors) in the wheel. The user will then rotate the wheel until blue is at position 4 before clicking 'confirm'.

If the user fails to authenticate themselves after 3 attempts, the account is locked. Additionally, after the user has successfully logged in 5 times, they would have to register again, rendering the information the shoulder surfer has obtained useless.

The scheme has a low chance of accidental login. However, the security of the scheme is difficult to quantify due to its complex nature that depends a lot on randomisation. Much of the scheme is left to chance, and the equations involving 'n' must be carefully crafted to maintain ambiguity to ensure that the value of 'n' cannot be figured out via analysis or cross checking equations.

Pass Rule Scheme

Another scheme that was considered was the Pass Rule Scheme (PRS), in which a user ranks 'x' number of pass rules in order from 1 to 'x' to use during login.

During registration, users select and rank the provided pass rules. The user then memorises the pass rules and their respective rankings.

During the login phase, the screen will show a pool of icons on the bottom and a number at the top of the interface, with the number indicating the ranking of the pass rule to be used during that round of the login process. According to the user's personal ranking of the pass rules, they have to recall their pass rule with the corresponding ranking and count the number of icons in the pool of icons that adhere to the corresponding pass rule. Afterwards, the user inputs their numerical answer into an entry box at the bottom of the interface. This is one round of authentication, the scheme continues for 3 rounds before the user is authenticated. E.g. pass rule is 'select animals', the pool of icons only contains 2 animals, a dog and a monkey, the user will then input 2 into the entry box.

However, it turns out that it was non-trivial to come up with these pass rules as they had to have overlap so as to introduce uncertainty, and could not be specific to certain icons, since it may reveal too much information about which pass rule was used during the recorded round of login. The observer could then be able to guess which pass rule was used and associate it to the ranking represented by the number displayed at the top of the recorded interface during login. The icons also had to be classified into as many different categories as possible, which proved to be a challenge.

Icons Scheme

A scheme we brainstormed was the Icons Scheme (IS), in which a user chooses icons and counts them to form a 4-digit PIN for login. A source of inspiration we had while coming up with this scheme was the Miller's Law which states that the number of objects an average person can hold in working memory is 7 ± 2 [8].

In the registration phase, the user selects 8 of 20 icons. The user then memorises the icons in no particular order. The selected icons would become the registered pass-icons of the user.

During the login phase of the scheme, there will be 4 panels displayed on the screen, each having a pool of different randomly generated icons, each appearing a specific number of times. Each panel would contain only 1 pass-icon, and some icons would have the same count as the pass-icon in order to introduce uncertainty and ensure the shoulder surfer is unable to outrightly guess the pass-icons. The user has to count the number of times their registered pass-icons appear in

each panel, and input that number into an entry box at the bottom of the interface to form a 4-digit PIN. The users will be authenticated if their PIN is correct.

However, the scheme requires the icons to be specifically generated with calculated randomness, as at least 2 icons in each panel need to be of the same count, so as not to easily give away which icon in the panel had been selected as the pass-icon of the user. For example, if only the star icon in one panel appears 3 times, and the user input the number '3' into his PIN to submit his response, then the observer can immediately figure out that the star is one of the registered pass-icons of the user. Additionally, the usability of the system may not be very high as users have to spend time searching for and counting their pass-icons, which is time-consuming and may frustrate certain users.

Pictures Scheme

Another scheme we had come up with was the Picture Scheme (PS). The user selects an alphabetical password, and logs in via pictures of objects.

In the registration phase, the user chooses an alphabetical password of up to 8 characters. They memorise their password as per usual, but do not directly use it to log into their account.

In the login phase, the interface displays the same number of panels as the length of the user's password. In each panel, the user will see multiple objects, of which the names of all are spelt with different starting characters (e.g. apple = "a", bagel = "b"). Each panel coincides with the position of each character of the password. The user clicks on the object with the same starting character as the character of their password. In each panel, there will only be one 'correct' object representing the alphabet being verified. With each selection, the selected object will be circled. After the user selects all their icons, they will tap on a button on the bottom right corner of the screen to submit their response. Once authenticated, the user will be able to login.

However, we felt that the pattern would be easily deciphered by observers after multiple observations, hence decreasing the security of the scheme. There is also a possibility that the users may not be able to recognise the objects accurately, (e.g. they mistake an apple for another fruit) as planned by the programmer.

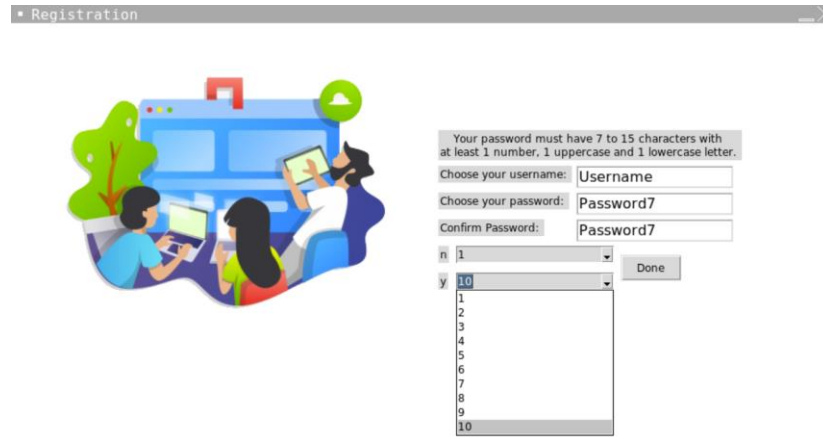
4. Design of the Proposed Scheme

In this section we propose a password scheme, the Character Based Verification Scheme (CBVS), that has quantifiable security and introduces uncertainty during the login phase of the password scheme, unlike a conventional alphanumeric authentication scheme. The alphabet used in the proposed scheme contains 64 characters, including 26 upper case letters, 26 lower case letters, 10 single digit numbers ; 0 to 9, and 2 symbols ; '@' and '!'.

A. Registration phase

1. The user selects a username, followed by an alphanumeric password with 7 to 15 characters, containing at least 1 uppercase letter, 1 lowercase letter and 1 numeral. The user also selects a value for 'n' and 'y' respectively. 'n' and 'y' are variables, each has a

value ranging between 1 to 10 and both will be used during the login phase. The values of ‘n’ and ‘y’ are known by the user only. An example is illustrated in Fig.1.



The registration form is titled "Registration". It includes an illustration of three people using laptops. The form fields are:

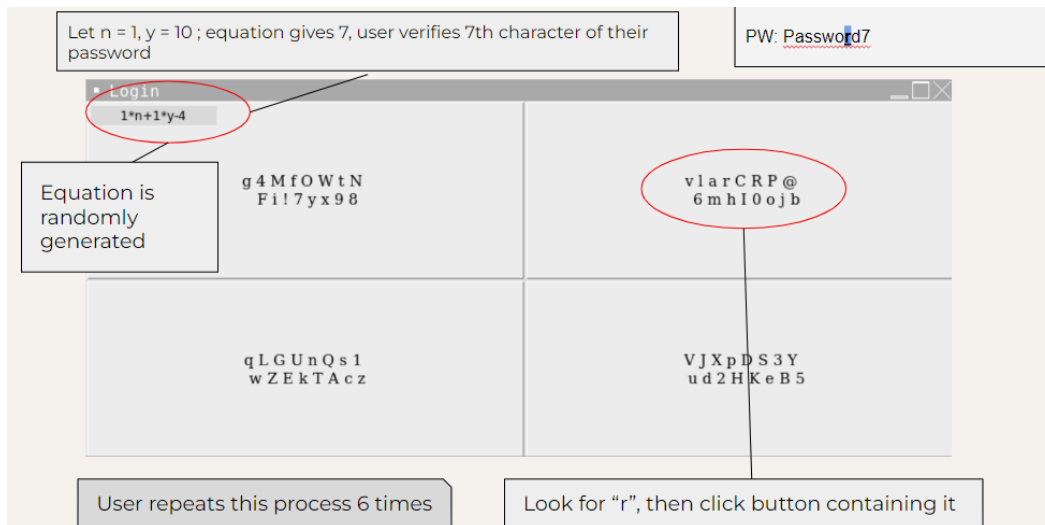
- Choose your username:
- Choose your password:
- Confirm Password:
- n:
- y:
- A "Done" button.

Below the 'y' field is a list of numbers from 1 to 10.

Fig.1: An example of a registration

B. Login phase

1. The user then proceeds to the login page to enter their username. Once the username of the user has been authenticated, the login process begins. The user will be shown an interface with an equation on the upper left corner of the screen. The equations will have the format of $A(n) \pm B(n) \pm C$, where A and B each have a value ranging from 1 to 3 while C has a value of 0 to 6. 4 buttons, each with 16 randomly generated characters of the 64 characters in the alphabet used in this proposed scheme (including the symbols “!” and “@”). The interface is displayed in Fig. 2



The login interface is titled "Login". It includes an illustration of a person using a laptop. The interface elements are:

- Equation: $1*n+1*y-4$ (circled in red)
- Equation explanation: "Equation is randomly generated"
- Equation result: "Let n = 1, y = 10 ; equation gives 7, user verifies 7th character of their password"
- Password field: "PW: Password7" (with the 7th character highlighted)
- Four buttons with 16 randomly generated characters each:
 - g4MfOWtN Fi!7yx98
 - vlarCRP@ 6mhI0oJB (circled in red)
 - qLGUnQs1 wZEKTAcz
 - VJXpDS3Y ud2HKeB5
- User instruction: "User repeats this process 6 times"
- User instruction: "Look for 'r', then click button containing it"

Fig.2: An example of a login interface based on Fig.1

2. The user has to calculate the value of the equation provided. The end value of the equation correlates to the position of the character in the user's password to be verified.

(e.g. end value= 7; Password = Password7; User clicks the button with the character “r” on it.)

3. Fig.2 shows an example of 1 round of verification. The full login process of the proposed scheme has a total of 6 rounds of verification with the same interface for all 6 rounds.
4. In order to introduce uncertainty into the proposed scheme, we decided to include a feature where users have to key in intentional wrong answers. The number of intentional wrong answers per login process ranges from 0 to 2. The intentional wrong answers do not affect the total number of rounds of verification, which is 6, so as to not let an observer know when an intentional wrong answer has been inputted.
5. Users are signalled to input an intentional wrong answer when the end value of the provided equation is ≤ 0 or $>$ the length of the password. E.g. (Password = Password7, end value of equation = 0 or 10). User keys in an intentional wrong answer by selecting any of the 4 buttons on the interface.
6. After the user successfully passes all 6 rounds of verification, the system authenticates them, and the user logs in. However, when the user gets a round of verification wrong, the system will not immediately terminate the login attempt. Only at the end of all 6 rounds, will the system display “Login Unsuccessful”. It takes one round of verification being done wrongly for the login attempt to be deemed unsuccessful. This is to introduce uncertainty as the observer will not know which round of verification the user, or the observer themselves, had gotten wrong. The user is given 2 chances to login.
7. After 3 successful logins from the user, the user has to change their password and value of ‘n’ and ‘y’.

The following section details the key space and accidental login rate of the scheme.

5. Analysis of the Proposed Scheme

5.1 Security of the proposed scheme

In this section, we will analyse the security and usability of the proposed scheme in terms of the worst case scenario, where the user uses a 7 character password.

A. Keyspace

The maximum keyspace of a 7 character long password prior to any observations = 64^7 or 2^{42} , since the general formula of the key space of an x character password is 2^{6x} .

B. Resistance to accidental login

Taking the worst case scenario where there are 2 intentional wrong answers to be inputted during login, Probability of a successful login = $(1/4)^4 = 1/256$ or 2^{-8} . Probability of getting at least one round of verification wrong = $255/256$. Since observer will have 2 login attempts, probability of accidental login with 2 attempts = $1/256 + 1/256 * 255/256 = 511/65536$ which is about 2^{-7} or 0.77972% success rate. If the number of intentional wrong answers in a login attempt was lesser, resistance to accidental login increases even further.

C. Resistance to shoulder surfing

For every login interface, such as Fig.2, observed, on average, the keyspace of the proposed scheme is divided by 4 as the observer sees the user select one button out of 4. Thus taking the worst case scenario of a user using a 7 character password, with a maximum keyspace of 2^{42} , the password scheme can withstand 21 observations of the login interface, which is 3.5 full login processes. Each login process has a $2/3$ chance of having 1 or 2 intentional wrong answers, thus 21 login interfaces could be more than 3.5 full login processes. Besides, for our proposed scheme, after 3 successful logins from the user, the user has to change their password as well as their selected values of 'n' and 'y'. Hence, although the observer needs 21 observations to recover the full user password, the password of the user would be reset after just 3 successful logins, hence the observers can get a maximum of only 2 recordings, which is insufficient to crack the scheme. Even if the user does not change their password after 3 successful logins, in the worst case scenario with no intentional wrong answers, the scheme can still withstand a minimum of 21 observations of the login interfaces. If users were to use a longer password, the scheme would withstand even more observations.

Additionally, that is assuming the observers are able to cross check their recordings and match recordings where the same characters are being verified, so as to gain any information about the password of the user. However, the characters of the password being verified each round are random as the provided equations are randomly generated based on the password length of users. Even if the observer were to take the 6 provided equations from each recorded full login process and cross check them to conduct guess and check to find out the values of 'n' and 'y', they need to know 2 things to do so: 1. Password length of user 2. Number of intentional wrong answers in the login process. The password length determines what end value of the equations would signal the user to input an intentional wrong answer. Not knowing the number of intentional wrong answers per recorded login process also makes it difficult for observers to carry out guess and check as they would have to assume the worst case scenario of there being 2 intentional wrong answers per recorded login process and password length being 15, hence leaving them with a greater number of possible combinations of values of 'n' and 'y'. The only way for the observer to reduce the number of possible combinations of 'n' and 'y' would be to substitute every possible combination of 'n' and 'y' values into the 18 recorded provided equations from their 2 recordings of 2 full login processes, and cancel out the combinations that led to there being 3 end values smaller than 1 and greater than 15. Even if they go through this tedious process, it will leave them with a great number of possible combinations as with 'n' and 'y' both having values ranging from 1-10 each, they have 10^2 total possible combinations of values.

Observers cannot skip this step either as without knowing 'n' and 'y', they cannot figure out what position of the character being verified in each round of verification, thus they cannot cross check their recordings, and hence they cannot gain any information about the password of the user. In addition to that, observers would not be able to figure out which input was an intentionally wrong answer as they do not know which equation led to an end value <1 or $>$ than the unknown password length of the user. The observer would then be left to guess the password length of the user, in which the observer has a probability of $1/9$ chance of getting right, which is relatively small considering how they have to use it to further guess the values of 'n' and 'y' which would have an even smaller probability than $1/9$ of the observer getting the values right.

Furthermore, the password requirement is 7 to 15 characters long, so the password would have additional characters that may not be involved in the 6 rounds of verification as the characters being selected to be verified in each login attempt are randomly selected.

Additionally, the addition of intentional wrong answers also serve to introduce even more uncertainty as the observer does not know which inputs were for intentional wrong answers. Hence, the observer cannot rule those inputs out and they would confuse the observers during their cross checking of their recordings, causing the accuracy of their conclusions to decrease.

5.2. Usability experiments of the proposed scheme

In order to analyse how usable our scheme is, we invited 16 participants to test out our prototype, and recorded how long they take to register and successfully log in, all in terms of seconds (s). Additionally, we also took note of their login success rates and prepared a short survey for them to partake in once they had successfully logged in.

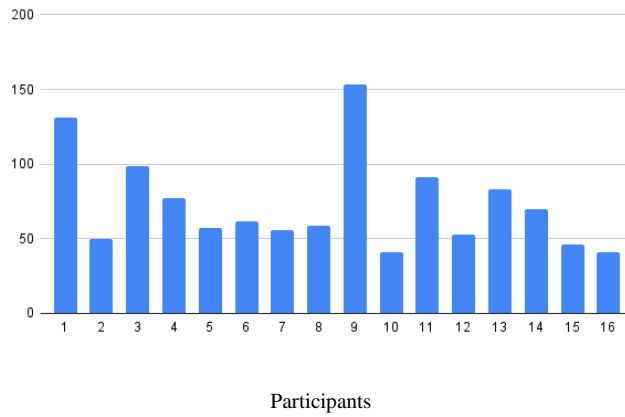


Fig.3 Time taken for each participant to successfully log in

According to Fig.3, the mean time taken for all participants to successfully log in was 73s. All participants had an average of 88% successful login rate. Additionally, we also had all participants repeat the login process multiple times with the same password and 'n' and 'y' values to see if the time they took to log in had any changes over time.

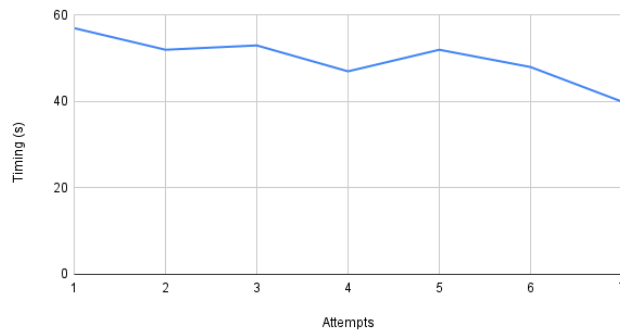


Fig.4 Mean time taken for successful logins of participants using the same password and value of 'n' and 'y' for all attempts

Our main concern for our scheme was that it requires users to carry out mental calculations to calculate the end values of the provided equations to know which character of their passwords to verify or whether or not to input an intentional wrong answer. We were afraid that this would hamper their login process and increase the time taken to successfully log in drastically.

However, Fig.4 shows that there is an overall decrease in the time taken for successful logins as the number of attempts increased. The password used was “Password7” and values of ‘n’ and ‘y’ used were 4 and 7 respectively. This shows that with more attempts, the participants got more familiar with the system and began to use less time to successfully log in.

Criteria	Perceived Security (1-10)	Ease of Use (1-10)	Efficiency (1-10)	Overall Satisfaction (1-10)
Mean Score	9.0	7.5	8.0	8.0

Table.1: Results of our survey

Table.1 shows the results of our survey. Perceived Security refers to how secure the participants felt using the proposed scheme and how secure they thought it was against shoulder surfing. Ease of Use refers to how easy it was to login with the proposed scheme and how easy it was to understand how the scheme works. Efficiency refers to how satisfied they were with the time they had to take to successfully login. Overall Satisfaction refers to how willing they would be to have to use this scheme again in the future to log into systems they use in their daily lives.

Some feedback given to us was that this system may not be welcomed by those who struggle with mathematics as they may be weak with mental calculations and cannot rely on a calculator as it would give away their chosen values of ‘n’ and ‘y’. Hence the calculations would hinder their time taken to successfully log in. Another surprising feedback given to us was that the participant actually found logging in with our system fun as it felt like a game where he could challenge himself to log in as quickly as possible. Additionally, since we let the participants choose their own passwords and values for ‘n’ and ‘y’, they had an easier time memorising them as compared to if we were to have used random generators instead.

6. Conclusion and Further Works

In conclusion, in this paper, we have proposed a scheme which has quantifiable security and is usable by the average human of the general public. The Character Based Verification Scheme authenticates the user by getting the user to verify characters in their password. The scheme can withstand a sufficient number of observations before the user resets their password. The scheme is also sufficiently resistant against brute force attacks and dictionary attacks. With this scheme, users can log in without worrying about shoulder surfing attacks. Future works should consider ways to reduce time needed to successfully login so as to increase usability of this proposed scheme even further.

7. Acknowledgements

The authors would like to thank their mentors Dr Sim Siang Meng and Mr Choo Jia Guang for their continuous support and advice throughout the project. This work was supported by Defence Science Organisation (DSO) National Laboratories in Singapore.

8. References

- [1] H. Zhao and X. Li, "S3PAS: A scalable shoulder-surfing resistant textual-graphical password authentication scheme," *Proc. of 21st Int.Conf.on Advanced Information Networking and Applications Workshops*, vol. 2, May 2007, pp. 467-472.
- [2] M. Sreelatha, M. Shashi, M. Anirudh, Md. Sultan Ahamer, and V. Manoj Kumar. "Authentication schemes for session passwords using color and images," *International Journal of Network Security & Its Applications*, vol. 3, no. 3, May 2011.
- [3] Jianwei Lai, Ernest Arko, 2021. A Shoulder-Surfing Resistant Scheme Embedded in Traditional Passwords. *Proceedings of the 54th Hawaii International Conference on System Sciences*. pp 7144-7152.
- [4] Lee, MK., Nam, H. (2013). Secure and Usable PIN-Entry Method with Shoulder-Surfing Resistance. In: Stephanidis, C. (eds) *HCI International 2013 - Posters' Extended Abstracts*. HCI 2013. *Communications in Computer and Information Science*, vol 374. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-39476-8_149
- [5] Gao, H., Liu, X., Wang, S., Liu, H. and Dai, R., 2009, December. Design and analysis of a graphical password scheme. In *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)* (pp. 675-678). IEEE.
- [6] Manjunath, G., Satheesh, K., Saranyadevi, C. and Nithya, M., 2014. Text-based shoulder surfing resistant graphical password scheme. *International Journal of Computer Science and Information Technologies*, 5(2), pp.2277-2280.
- [7] M Sreelatha, M Shashi, M Anirudh, MD Sultan Ahamer, V Manoj Kumar, May 2011. Authentication Schemes for Session Passwords using Color and Images. *International Journal of Network Security & Its Applications*, pp.115-117.
- [8] Miller, E.K. and Desimone, R., 1994. Parallel neuronal mechanisms for short-term memory. *Science*, pp.520-522.