Published *online* on the journal's web page : http://innovatics.unsil.ac.id

Innovation in Research of Informatics (INNOVATICS)

| ISSN (Online) 2656-8993 |

An Algorithm for Color-Based Password Authentication to Increase Security Level

Siti Rahayu Selamat[1], *Soung Young Cai*[2*], *Nor Hafeizah Hassan*[3], *Robiah Yusof* [4]

[1,3,4]*Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Malaysia*
[2]*Merimen Online Sdn Bh, Block D, Level 1, UPM-MTDC Technology Center III, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia*
[1]*sitirahayu@utem.edu.my*, [2]*ycsoung@merimen.com*, [3]*nor_hafeizah@utem.edu.my*, [4]*robiah@utem.edu.my*

ARTICLE INFORMATION

Article History:
Received: March 5, 2024
Last Revision: April 12, 2024
Published *Online*: April 13, 2024

KEYWORDS

Air Pollution,
Authentication,
Color-Based Algorithm,
Encryption,
Password

CORRESPONDENCE

Phone: +601812873542

E-mail: ycsoung@merimen.com

ABSTRACT

Security level in authentication is essential to decrease the possibility of an account being guessed. Several authentication methods are widely used nowadays, covering digital aspects such as passwords, challenge-response, public and private key / digital certificates, and physical elements such as fingerprints, iris, or retina scanning. This paper aims to focus on solving the problem of the password. This textual authentication consists of many vulnerabilities open to attacks like eavesdropping, dictionary attack, and brute force attack by increasing the level of complexity in the authentication algorithm. In this paper, we proposed a new color-based password authentication algorithm to solve the vulnerabilities in textual authentication. The color-based password authentication algorithm consists of three main processes: color selection, hexadecimal password encryption, and password verification. This research contributes to a new color-based authentication by increasing the complexity of the verification process that can solve the vulnerabilities of textual authentication and harden the level of security in the authentication layer. This color-based authentication algorithm could fully replace textual authentication in the future and is worth using in sensitive data domains such as medical and health or banking institutions.

1. INTRODUCTION

The complexity of the authentication algorithm plays a significant role in avoiding unwelcome access. Frequently, complexity refers to the length of a password used. However, increasing the password length has its problem, such as users could easily forget a password or use easily guessed passwords. Hence, this issue is getting attention from the public to prevent account access.

An algorithm for authentication using a colour mechanism is a new authentication method that can replace the current textual authentication. The existing textual authentication is vulnerable to many types of attacks such as eavesdropping, visual hacking, social engineering, dictionary attack, brute force attack, and others [1] [2]. This authentication will use its algorithm to store the data and validate it when the user login. This authentication will increase the complexity of the password compared to textual passwords by putting extra effort into the attackers to memorize their chosen password to provide maximum security.

This paper is organized as follows: First, we present state of the art in section 2. Next, we describe the design of the proposed algorithm in section 3. Subsequently, we detail the proposed colour-based authentication algorithm in section 4. Next is the evaluation in section 5 and the result and discussion in section 6. Finally, we give conclusions and future works in section 7.

2. RELATED WORK

Most of us are familiar with the textual password or plaintext authentication, which is the most common and widely used nowadays, yet it is still considered an insecure authentication method [3]. Furthermore, according to [4], password authentication is still ubiquitous, even though alternatives have been created to address its drawbacks, such as the significant cognitive load it places on users. To log onto a computer or network, you enter a user account name and the password assigned to that account. This password is checked against a database that contains all authorized users and their passwords. Since 1992, the observation of password change choice, strengths, and reusable was studied. For example, a study by [5] observed password change choices on 54 machines representing approximately seven thousand user accounts, as shown in Table 1.

TABLE 1: GENERAL MEANS OF AUTHENTICATION

Length	Number	Percentage (%)
1	55	0.4
2	87	0.6
3	212	2.0
4	449	3.0
5	1260	9.0
6	3035	22.0
7	2917	21.0
8	5772	42.0
Total	**13787**	**100.0**

Table 1 shows the observed password length conducted by Purdue University. The results showed that almost 3% of passwords were three characters or fewer, enabling the attacker to begin the brute force attack by exhaustively testing all possible combinations of passwords on the victim. It causes the account to be easily compromised, leading to information leakage.

The finding from Purdue University's study was supported by other results from [6], [7], and [8]. Their study shows that three out of four consumers use duplicate passwords, many of which have not been changed in five years or more. About 40 percent of those surveyed say they had an account hacked, password stolen, or were given notice that their personal information had been compromised. The studies also show that the top five popular passwords used by users in the year 2014 are "123456", "password", "12345", "12345678", and "qwerty," which are significantly easy to be guessed and cracked. It also created a 'domino effect' that allowed hackers to take down multiple accounts by breaking a single password. Other than that, several surveys conducted to identify the worst passwords used, as stated in [9] and [10], give offenders opportunities to attack the system.

In addition, [1], [11], and [12] reported that to preserve the security of the network, passwords must be “strong,” that is, they should contain a combination of alpha and numeric characters and symbols, they should not be words that are found in a dictionary, and they should be relatively long. In short, they should not be easily guessed. Table 2 shows some characteristics of a strong password.

Table 2 shows the characteristics and description of the strong password, which most users nowadays ignore. This issue has led to password authentication being vulnerable to a password “cracker” who uses a brute force attack that tries every possible combination until hitting upon the right one or who operates a protocol “sniffer” to capture packets if passwords are not encrypted when they are sent over the network. Hence, textual password authentication should be improved by a new method to strengthen or replace it.

TABLE 2: CHARACTERISTICS OF STRONG PASSWORD

No	Characteristic	Description
1	Minimum 8 Characters	Passwords should at minimum of 8 to 14 characters in length. A longer password would be even better which hardly being intruded.
2	Includes Numbers, Symbols, Capital Letters, and Lower-Case Letters:	Use a mix of different types of characters to make the password harder to crack.
3	Not a Dictionary Word or Combination of Dictionary Words:	Prevent usage of obvious dictionary words and combinations of dictionary words. Any combination of a few words, especially if they’re obvious, is also bad. For example, “house” is a terrible password. “Red house” is also very bad.
4	Doesn’t Rely on Obvious Substitutions	Don’t use common substitutions, for example, “l00p” isn’t strong just because you’ve replaced an o with a 0.

Cryptography technique can be defined as scrambling the data, which can protect the information from being readable by an adversary even when the data reaches them. This scrambling process also can be known as hidden writing to transform information into a secure form so that it can transmit or be stored safely from access by unauthorized persons [13]. There are a few cryptography techniques: a) classical cipher, b) substitution cipher, c) transposition cipher and d) modern cipher.

Classical cipher is an old technique that is easy to break into nowadays and is considered an insecure cryptography technique. For example, the Caesar cipher, a simple alphabetic shifter used in Roman times, is also based on a classical cipher. This cipher has a small essential space and can be easily broken by performing a brute force attack. This technique is the basis for the later ciphers.

A substitution cipher is also known as confusion by replacing parts of the letters with symbols or others, also known as meaningless data bits. This long-abandoned cryptographic method is still used as a framework for cryptography nowadays. Many cryptographic algorithms implement substitution cipher in their algorithms, such as Caesar Cipher, Hill Cipher, Monoalphabetic Cipher, Polyalphabetic Ciphers, and others [14] [15]. In this study, Caesar Cipher is selected and will be discussed in this section. Similar to a substitution cipher, a transposition cipher is one of some basic methods to produce or create a new secret code.

A transposition cipher is one of the methods which performs encryption by rearranging positions held by units of plaintext to some permutation which is its secret keys [16]. In other means, the order of units of the plaintext is reordered in position and creates a new unique secret code. Many famous cipher algorithms use transposition as their frameworks, such as Rail Fence cipher, Route Cipher, Columnar transposition, double transposition, Myszkowski transposition, and others [17].

Modern cipher is heavily based on computer science practices and mathematical theory to design an algorithm which hard to break in from reaching adversaries [18]. One of the famous modern ciphers widely used is Advanced Encryption Standard (AES), which replaces Data Encryption Standard (DES).

In modern cryptography, many algorithms are based on a mathematical calculation, which significantly improves from the classical cipher. However, most modern ciphers still adapt the concept of the classical cipher by using substitution and transposition in their algorithm, such as Advanced Encryption Standard (AES) [19]. Plaintext data should always be encrypted, especially passwords, which is an essential part of authentication to access users' private information. Applying cryptography to the data may not necessarily entirely or wholly secure the data, but it can slow down the adversary's encryption if they wish to break it.

3. Methodology

This study produced a new authentication method, colour-based authentication, combined with a new algorithm that uses a cryptography approach to encrypt and decrypt the data. A colour chart with different colours and hexadecimal values will be displayed to the user to authenticate as a password. The base cryptography algorithms used in this study are the concept of substitution (Caesar Cipher), transposition cipher (Rail Fence cipher), and Advanced Encryption Standard (AES) to create a unique secret code before being stored in the database, which can ensure the security of the password.

A. Caesar Cipher

Caesar cipher is one of the first ciphers ever created by Julius Caesar. This cipher uses using substitution algorithm in implementation by rearranging the alphabet shifting three spaces to the left as in Fig. 1. Standard alphabet order starts with the alphabet a/A, but in Caesar cipher, the first alphabet will be d/D which had been shifted three spaces to the left.

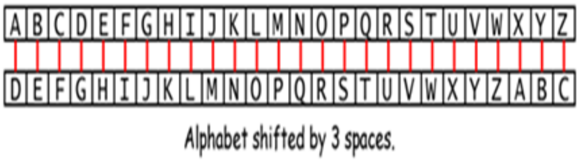

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
Alphabet shifted by 3 spaces.

Fig. 1 Example of Caesar Cipher

As an example, the plaintext of ENCRYPT will become HQFUBSW after performing Caesar cipher on it. Due to the simplicity of this shifting technique, this is considered a basic form of encryption, which is extremely easy to break by using brute-force attacks. Hence, the Caesar cipher's simplicity caused this algorithm to no longer be used, and the substitution cipher is used as the base of cryptography to produce a more complex algorithm [20].

B. Rail Fence Cipher

A Rail Fence cipher is a cipher that uses a zig-zag technique [21]. In Fig. 2, the cipher produced will be "CMHMTMROOEOEOOEW". The algorithm is implemented by using the concept of zig-zag. The plaintext is written downwards and diagonally on successive "rails" of an imaginary fence, then will move up until it reaches the bottom rail.

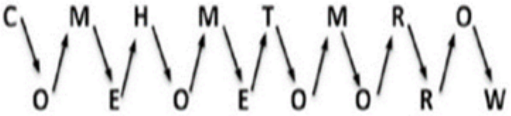


Fig. 2 Example of Rail Fence Algorithm

The message will be read in rows and produce a new secret code. This algorithm is also not a strong one that is vulnerable to many attacks. However, the principle of transposition is still used in cryptography which can produce a better algorithm by increasing the complexity of transposition and substitution [22].

C. Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES), also known as Rijndael, is chosen to represent the group of modern ciphers due to its relatively new product cipher [23]. AES was developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen. AES was designed to solve and overcome the flaws in the design of DES, which has become a new replacement for DES and triple DES. AES is an algebraic algorithm with a simple mathematical structure, but it's not easy to break. AES is based on a design principle that combines substitution and permutation, which is fast in software and hardware. Only two attacks are successful on AES: "Square attacks" and algebraic attacks on the S-box [24]. The structure of the AES will be shown in Fig. 3.

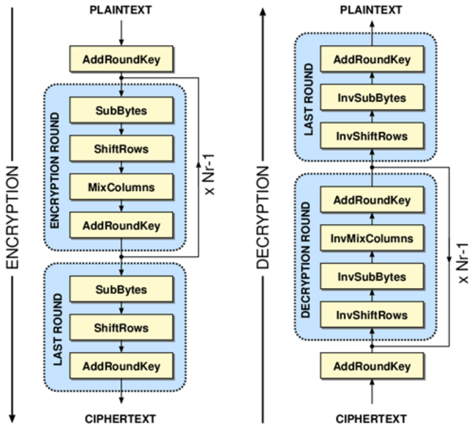


Fig. 3 Structure of AES (Nvarko-Boateng et al., 2017)

Fig. 3 shows the structure design of the AES. The AES is designed using four main steps; a) "SubBytes" step (each byte is replaced using a lookup table); b) "ShiftRows" step (transposition step which shifted a certain number of steps on the last three rows of the states cyclically); c) "MixColumns" step (mixing operation by combining four bytes in each column of the state); and d) "AddRoundKey" step (bitwise XOR to combine each byte of the state with a block of a round key) [25].

The design and strength of the AES algorithm of all key lengths are sufficient to protect classified information up to a SECRET level, which requires at least 192-bit keys at a minimum. The design of AES, which adapt the concept of classical cipher, is also widely used by the public as a reference to create their cryptography algorithm.

4. Proposed Color-Based Authentication Algorithm

This section describes the algorithm used in this study: the combination of substitution cipher, transposition cipher, and AES cipher techniques. In colour-based authentication, a user must submit two things: a) username and b) a preferred colour. Each colour selection input through the colour chart produced six hexadecimal characters. Users must select a minimum of six colour inputs and up to 16 color inputs as their password, which will create a string. This string's size can range from at least 36 hexadecimal characters up to 96 hexadecimal characters. Each of these characters is stored as an individual element of an expandable 3x3-based array. For example, a six-colour password will produce 36 hexadecimal characters and be stored in a two-dimensional array of size 6x6. This color is encrypted from hexadecimal characters into decimal secret code using a new color-based authentication algorithm and kept in a database. Later, to verify a user's attempt at an account, it will be decrypted and compared.

A. Colour-based Encryption Algorithm

Encryption is the process of changing or altering information, especially passwords, to make it unreadable by anyone except those possessing special knowledge, which is a "key" that enables them to convert the data back to the original and readable form. The first process of the encryption module starts when the hexadecimal password is converted to decimal. Every 3rd and 4th character will be converted from hexadecimal into decimal before inserting into the 3x3 matrix. The first color selection will be used to illustrate the arrangement of the converted password on the 3x3 matrix.

Algorithm 1: Convert Hexadecimal Password to Decima

```
$A_c$ = Array Cell
$H_p$ = Hexadecimal Password
$D_p$ = Decimal Password
$D_pL$ = Decimal Password Length
1.0 Start
2.0 If ($A_c$ == 2 && $A_c$ == 3)
        2.1 Convert from $H_p$ → $D_p$
3.0 If ($D_pL$ == 1)
        3.1 Insert $D_p$ into $A_c$[4]
        3.2 Insert 1 → $A_c$[5] && $A_c$[6]
        3.2 Else If ($D_pL$ == 2)
            3.2.1 Insert 1 → $A_c$[6]
        3.3 Else
            3.3.1 Insert NULL into $A_c$[3]
            3.3.2 Insert $D_p$ into $A_c$[4]
            3.3.3 Insert NULL into $A_c$[7]
4.0 Insert NEXT $H_p$ → $A_c$[8]
5.0 End
```

Fig. 4 Convert Hexadecimal Password to Decimal Algorithm

Based on Algorithm 1, shown in Fig. 4, if the position is equal to 3rd and 4th characters or array column equal to 2 and 3, the hexadecimal value will be converted from hexadecimal into decimal. If the conversion's output from hexadecimal to decimal is only 1 or 2 digits, a "1" will be inserted at the end to complete it into three numbers using the algorithm.

The result of the seven-character password will be further processed using the converted password duplication method to fill up the space in the 3x3 matrix, which will end with a nine-character password. The converted password duplication phase will duplicate from every second row first column converted password into space in third row first column and every second-row third column converted password into space in the first-row third column. This process is represented in Algorithm 1, and the converted password duplication of the sample data is shown in Fig. 5.

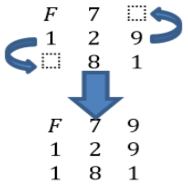


Fig. 5 Example converted password duplication process on data

After all the matrices are filled with the converted password from the previous process, the hashing will start with reflection or mirroring to alter the position of the password. This phase will reflect the converted password in the first column of every row into the third column of every row, and conversely, the third column of every row will be reflected into the first column of every row. The second column of every row will remain unaltered at the end of the process. The reflection or mirroring of the converted password from Fig. 5 is shown in Fig. 6, and the mirroring process is presented in Algorithm 2, as shown in Fig. 7.

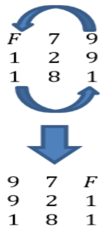


Fig. 6 Example of mirroring or reflection process on sample data

Algorithm 2: Mirroring or Reflection the Converted Password

```
$A_c$ = Array Cell
$V_{cp}$ = Value of converted password
$V_{tmp}$ = Temporary Variable
1.0 Start
2.0 If ($A_c$ == 2)
        2.1 Store $V_{cp}$($A_c$[0]) → $V_{tmp}$
        2.2 Insert $V_{cp}$($A_c$[0])) → $A_c$[2]
        2.3 Insert $V_{cp}$($V_{tmp}$) → $A_c$[0]
3.0 If ($A_c$ == 5)
        3.1 Store $V_{cp}$($A_c$[3]) → $V_{tmp}$
        3.2 Insert $V_{cp}$($A_c$[3]) → $A_c$[5]
        3.3 Insert $V_{cp}$($V_{tmp}$) → $A_c$[3]
4.0 If ($A_c$ == 8)
        3.1 Store $V_{cp}$($A_c$[6]) → $V_{tmp}$
        3.2 Insert $V_{cp}$($A_c$[6]) → $A_c$[8]
        3.3 Insert $V_{cp}$($V_{tmp}$) → $A_c$[6]
5.0 End
```

Fig. 7 Mirroring or Reflection the Converted Password Algorithm

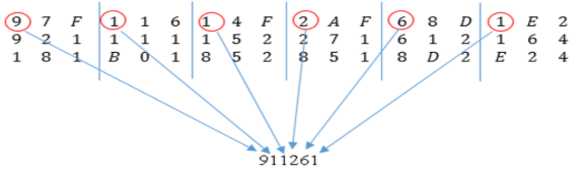
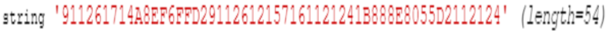

9 7 F 1 1 6 1 4 F 2 A F 6 8 D 1 E 2
9 2 1 1 1 1 1 5 2 2 7 1 6 1 2 1 6 4
1 8 1 B 0 1 8 5 2 8 5 1 8 D 2 E 2 4
911261

Fig. 8 Example of output by transposition of every first data in different matrix

Based on Fig. 8, the converted password will be combined continuously until the end of the matrix by using implode function to completely alter the converted password position and convert the converted password from a multidimensional array into the string format. All converted strings in nine variables will be joined or combined, forming a unique string. The resulting output by the transposition on all six color selections is shown in Fig. 9, and the algorithm used is presented in Algorithm 3, as shown in Fig. 10.



```
string '911261714A8EF6FFD29112612157161121241B888E8055D2112124' (length=54)
```

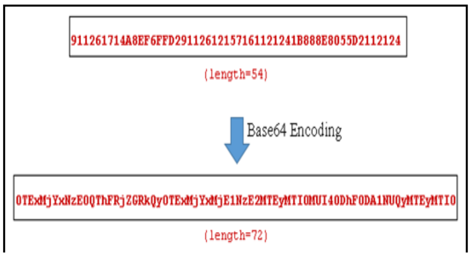
Fig. 9 Output of sample data after transposition process

Algorithm 3: Transposition the Converted Password

```
$A_c$ = Array Column
$V_{cp}$ = Value of converted password
$A_r$ = Array Row
V = Variable
$S_c$ = Combined String
1.0 Start
2.0 While ($A_r \leq 8$ && $A_c$ == 0)
        2.1 Insert $V_{cp}(A_r[i]A_c[0]) \rightarrow V_1$
        2.2 i++
3.0 While ($A_r \leq 8$ && $A_c$ == 1)
        3.1 Insert $V_{cp}(A_r[i]A_c[1]) \rightarrow V_2$
        3.2 i++
4.0 While ($A_r \leq 8$ && $A_c$ == 2)
        4.1 Insert $V_{cp}(A_r[i]A_c[2]) \rightarrow V_3$
        4.2 i++
5.0 While ($A_r \leq 8$ && $A_c$ == 3)
        5.1 Insert $V_{cp}(A_r[i]A_c[3]) \rightarrow V_4$
        5.2 i++
6.0 While ($A_r \leq 8$ && $A_c$ == 4)
        6.1 Insert $V_{cp}(A_r[i]A_c[4]) \rightarrow V_5$
        6.2 i++
7.0 While ($A_r \leq 8$ && $A_c$ == 5)
        7.1 Insert $V_{cp}(A_r[i]A_c[5]) \rightarrow V_6$
        7.2 i++
8.0 While ($A_r \leq 8$ && $A_c$ == 6)
        8.1 Insert $V_{cp}(A_r[i]A_c[6]) \rightarrow V_7$
            8.2 i++
9.0 While (Ar $\leq 8$ && $A_c$ == 7)
            9.1 Insert $V_{cp}$(Ar[i]$A_c$[7]) $\rightarrow V_8$
            9.2 i++
10.0 While (Ar $\leq 8$ && $A_c$ == 8)
            10.1 Insert $V_{cp}(A_r[i]A_c[8]) \rightarrow V_9$
            10.2 i++
11.0 Combine $V_1$, $V_2$, $V_3$, $V_4$, $V_5$, $V_6$, $V_7$, $V_8$, $V_9$
     $\rightarrow S_c$
12.0 End
```

Fig. 10 Transposition the converted password Algorithm

The output of the transposition phase will continue to alter by using encoding. In this encryption algorithm, base64 encoding is selected to be used to encode the converted password produced from the transposition phase. Base64 encoding will hash the converted password and increase the length of the converted password slightly from the previous process. An original output of the converted password will be combined with a reversed in every single character to form a new string. This process will further increase the complexity of the converted password, which will cause the attacker hard to find out the algorithm of the encryption process. The data encoding process using base64 encoding on sample data is shown in Fig. 11, and the algorithm for encoding the converted password is presented as Algorithm 4 in Fig. 12.



```
911261714A8EF6FFD29112612157161121241B888E8055D2112124
(length=54)
```

```
0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE2MTEyMTI0MUI40DhF0DA1NUQyMTEyMTI0
(length=72)
```

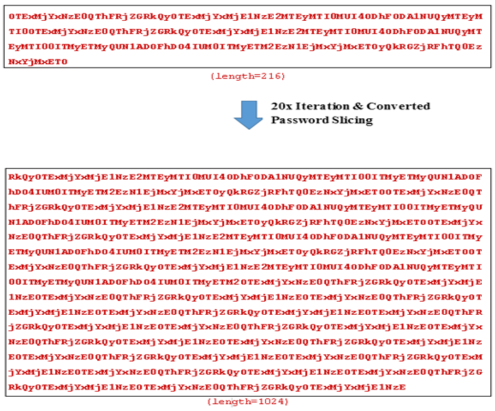
Fig. 11 Output of sample data after base64 encoding.

Algorithm 4: Encode the Converted Password

```
$B_{64}$ = Base64 Encoder
$V_{cp}$ = Value of converted password
V = Variable
1.0 Start
2.0 Encode converted password using $B_{64}$
3.0 Reverse $V_{cp}(B_{64}) \rightarrow V_1$
4.0 Combine Vcp & Vcp($B_{64}$) $\rightarrow$ V2
5.0 End
```

Fig. 12 Encode the Converted Password

Lastly, with its reverse string, the output of the transposition process will be passed to the iterating process. The process will be repeated 20 times, with each iteration returning a different converted password and the data being produced into a long string, resulting in extensive hashed data. The converted password will be sliced to make a string of 1024 characters in every iteration of the converted password slicing phase. Fig. 13 shows the iteration process's converted password slicing procedure on sample data, while the algorithm for this process is presented as Algorithm 5 in Fig. 14.



```
0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE2MTEyMTI0MUI40DhF0DA1NUQyMTEyM
TI00TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE2MTEyMTI0MUI40DhF0DA1NUQyMT
EyMTI00ITMyETMyQUN1AD0FhD04IUM0ITMyETM2EzN1EjMxYjMxET0yQkRGZjRFhTQ0Ez
NxYjMxET0
(length=216)
```

```
RkQy0TExMjYxMjE1NzE2MTEyMTI0MUI40DhF0DA1NUQyMTEyMTI00ITMyETMyQUN1AD0F
hD04IUM0ITMyETM2EzN1EjMxYjMxET0yQkRGZjRFhTQ0EzNxYjMxET00TExMjYxNzE0QT
hFRjZGRkQy0TExMjYxMjE1NzE2MTEyMTI0MUI40DhF0DA1NUQyMTEyMTI00ITMyETMyQU
N1AD0FhD04IUM0ITMyETM2EzN1EjMxYjMxET0yQkRGZjRFhTQ0EzNxYjMxET00TExMjYx
NzE0QThFRjZGRkQy0TExMjYxMjE1NzE2MTEyMTI0MUI40DhF0DA1NUQyMTEyMTI00ITMy
ETMyQUN1AD0FhD04IUM0ITMyETM2EzN1EjMxYjMxET0yQkRGZjRFhTQ0EzNxYjMxET00T
ExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE2MTEyMTI0MUI40DhF0DA1NUQyMTEyMTI
00ITMyETMyQUN1AD0FhD04IUM0ITMyETM20TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE
1NzE0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE0TExMjYxNzE0QThFRjZGRkQy0T
ExMjYxMjE1NzE0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE0TExMjYxNzE0QThFR
jZGRkQy0TExMjYxMjE1NzE0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE0TExMjYx
NzE0QThFRjZGRkQy0TExMjYxMjE1NzE0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1Nz
E0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE0TExMjYxNzE0QThFRjZGRkQy0TExM
jYxMjE1NzE0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE0TExMjYxNzE0QThFRjZG
RkQy0TExMjYxMjE1NzE0TExMjYxNzE0QThFRjZGRkQy0TExMjYxMjE1NzE
(length=1024)
```

Fig. 13 Final output of encrypted sample data

Algorithm 5: Slicing the Converted Password & Store the Encrypted Password

```
$L_n$ = Number of Loop
$V_{cp}$ = Value of converted password
SL = String Length
$S_o$ = Original String
$S_n$ = New String
V = Variable
1.0 Start
2.0 While ($L_n \leq 20$)
        2.1 Combine $V_{cp}(S_o)$ & $V_{cp}(S_n)$
        2.2 If SL > 1024 characters
        2.3 Slice the data using algorithm
        2.4 Else
        2.5 Proceed to next iteration
3.0 Store $S_n \rightarrow$ V
4.0 End
```

Fig. 14 Slicing the Converted Password & Store the Encrypted Password

Based on Fig. 13, the final output from the converted password through the last process, which is then the converted password slicing process, will produce the encrypted password. An encrypted password will contain 1024 characters after undergoing an encryption module using the converted password. The encrypted password will have all the possible combinations of characters such as uppercase letters, lowercase letters, and integers.

The result is based on the usability of the algorithm by entering the input required. To test the usability of the color-based authentication algorithm, which uses a key logger to capture the information for security testing (ICT).

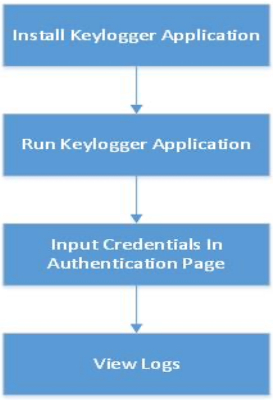
B. *Input Capture Testing*

Keystroke recording is the most common and effective attack method by hackers nowadays, which only requires a keylogger application that is widely available for download around the internet. The hacker will install the keylogger program inside the user's pc and hide it from discovery by the victim. Some keyloggers are easier to use, which requires the hacker to plug in the pen drive to the victim and run the execution file to instant run the keylogger without any installation.

Keystroke recording, which can capture the input from the victim's keyboard, will cause the victim to lose their private information and login credentials, leading to a severe impact on the victim. As the keystroke recording directly captures the input from the victim keyboard, any encryption used by the website, such as a secure sockets layer (SSL), cannot protect the user from losing their private information and login credentials. The process of conducting input capture testing is shown in Fig. 15.

In this input capture testing section, an open source keylogger application will be used to test the color mechanism authentication algorithm, which aims to solve the issues of keystrokes recording attacks on the victim's computer. The input capture testing will show the difference between textual authentications, which are commonly used, and color mechanism authentication to input the password.

The free keylogger that can be downloaded from the internet will be run or executed to start input capture testing. Inside the free keylogger application, click on the start capture keystrokes button to capture the input from the keyboard. After launching the keystrokes recording function, navigate to the textual authentication login page to insert your username and password. Both input fields are input through the keyboard.


Install Keylogger Application

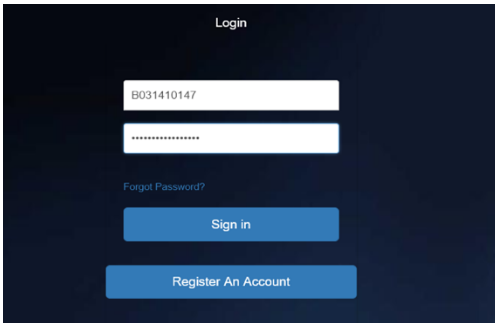
Run Keylogger Application

Input Credentials In Authentication Page

View Logs

Fig. 15 Input Capture Testing Process

The information will be captured as the computer's keylogger application runs. The username and password are input into the textual authentication login page, as shown in Fig. 16.

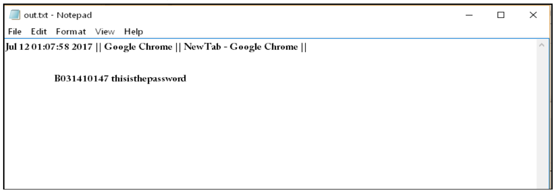

Login

B031410147

••••••••••••••••••

Fig. 16 Insert login credential through textual authentication.

After inputting the login credential into the textual authentication login page, navigate to the free keylogger application to stop capturing keystrokes and view the logs. The result of the keystroke recording is displayed inside the logs file, which will capture the date, application name, and the data that victims input through their keyboard. The login credential of the input through the keyboard, which consists of username "B03141014" and password "thisisthepassword", are shown in raw data form in the log file as shown in Fig. 17.

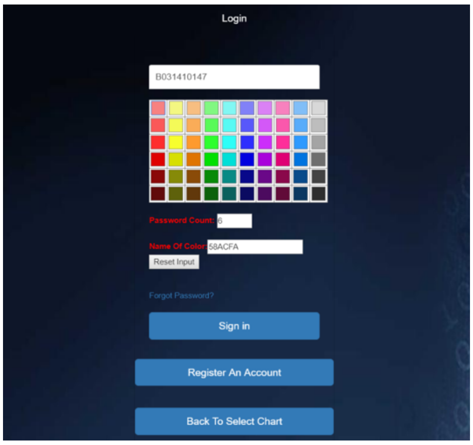


```
Jul 12 01:07:58 2017 || Google Chrome || NewTab - Google Chrome ||

          B031410147 thisisthepassword
```

Fig. 17 Output of keystrokes recording on textual authentication.

The input capture testing will be continued by conducting testing on the color mechanism authentication algorithm developed in this project. Like the previous example, repeat the process of start launch the keystrokes recording in the free keylogger application. After launching the keystrokes recording function, navigate to the login page, which uses the color mechanism authentication algorithm to input the login credentials, which also consist of username and password, as shown in Fig. 18.


Login

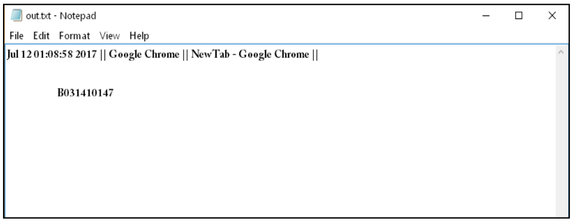
B031410147

Password Count: 6

Name Of Color: 58ACFA

Fig. 18 Insert login credential through color mechanism authentication algorithm

After inputting the login credential into the color mechanism authentication login page, repeat the process of navigating to the free keylogger application to stop capturing keystrokes and view the logs. The log file will display the login credentials inputted by the end user through the color mechanism authentication login page. The keylogger can only capture the end user's username, and the login credential's password cannot be captured, as shown in Fig. 19.



```
Jul 12 01:08:58 2017 || Google Chrome || NewTab - Google Chrome ||

        B031410147
```

Fig. 19 Output of keystrokes recording on color mechanism authentication.

Input capture testing has shown that the color mechanism authentication algorithm provides more secure user protection than textual authentication.

5. Result And Discussion

Based on the testing explained in the previous section, the result of input capture testing is summarized in Table 3.

TABLE 3: COMPARISON OF TEXTUAL AUTHENTICATION AND COLOR MECHANISM AUTHENTICATION ALGORITHM

Authentication Method	Characteristics		
	Input/Output devices required	Keystrokes recording	Login Credentials Captured
Textual Authentication	Keyboard	Vulnerable	Both of username and password of victim
Color Mechanism Authentication Algorithm	Keyboard and mouse	Invulnerable	Username of victim only

Table 3 shows that the textual authentication method is highly vulnerable to keystrokes recording attacks using a keylogger. Conversely, the color mechanism authentication algorithm, including username input through the keyboard and password input through color selection using the mouse, is invulnerable to keystrokes recording attacks using a keylogger. The color mechanism authentication algorithm provides higher security protection to end users than textual authentication. The color-based authentication algorithm is compared with textual authentication as in Table 4.

TABLE 4 COMPARISON BETWEEN COLOR –BASED AUTHENTICATION ALGORITHM WITH TEXTUAL AUTHENTICATION

Characteristics	**Method**	
	Color Mechanism Authentication Algorithm	**Textual Authentication**
More Secure	√	×
More User Friendly	×	√
More Invulnerable Against Social Engineering Attack	√	×
More Invulnerable Against Visual Hacking Attack	√	×
More Harder To Share	√	×
More Easier To Memorize	×	√

Table 4 shows that the color mechanism is more secure than textual authentication. The color mechanism authentication algorithm has significantly provided better protection to users as the color mechanism authentication algorithm is invulnerable to input capture attacks which often use keylogger as the tool to capture the keyboard input from the victims.

However, there will always be a significant tradeoff between security and user-friendly. The color mechanism authentication algorithm, which focuses more on protecting the user credentials or password in their authentication, has also reduced the user-friendly authentication progress as users need to recognize and search for a color to input the password through the color chart. Furthermore, the color mechanism authentication algorithm is also unsuitable for color blindness users to use to authenticate their identity as color mechanism authentication algorithms require users to select their colors. The color blindness users can only authenticate their identity by memorizing the color hexadecimal value, and inputting their password has caused their authentication process becomes complicated and thus reduced the user friendly.

The color mechanism authentication algorithm is also more invulnerable to some of the most common attacks, such as social engineering attacks and visual hacking attacks. The complexity and similarity of color caused the password to be harder to share and more challenging to memorize than textual authentication. These characteristics help the users to protect their passwords by providing higher security against various attacks.

The color mechanism authentication algorithm has a significant advantage over textual authentication in terms of security level. The color mechanism, which is invulnerable to input capture attacks, has solved one of the major issues nowadays: attackers can easily capture input by using a keylogger.

6. CONCLUSIONS

The colour chart for colour selection is available to be configured by the administrator to ensure the color chart able to suit to security level and interface design needed by the organization or developers. The one-way encryption also produces a long and complex hashing password to ensure the password is difficult to decrypt even if the attacker or adversary successfully steals the password through sniffing or other possible attacks on the password. The encrypted password will be produced into a long-length password of 1024 characters to resist attacks such as brute force attacks. In the next chapter, the colour mechanism authentication algorithm will be tested to ensure its reliability and effectiveness.

The final product or output from this study will solve many issues in textual authentication and harden the level of security in the authentication layer. This study has produced a novel authentication mechanism by using a color that may be able to replace textual authentication in the future. It will improve the authentication layer in any website that is accessible and highly vulnerable to attack through various hacking methods such as a brute force attack, dictionary attack, and others. This study is suitable even for system application by applying the concept to the development process. Authentication using a color mechanism can also help users protect their passwords. It is less likely vulnerable to attacks such as visual hacking and password guessing as authentication through color selection is more complex and hardly related to the user information than textual authentication, in which most users use their personal information in the password.

REFERENCES

Lal, N. A., Prasad, S., and Farik, M., “A Review of Authentication Methods”, International Journal of Scientific & Technology Research, vol. 5, no. 11, pp. 246-249, 2016.

Awan, K. A., Ud Din, I., Almogren, A., Kumar, N., and Almogren, A., “A Taxonomy of Multimedia-based Graphical User Authentication for Green Internet of Things”, ACM Transactions on Internet Technology (TOIT), vol. 22, no. 2, pp. 1-28, 2021.

Gould, E. M, “Authentication Methods and Recent Developments”, Serials Review, vol. 44, no. 3, pp. 247-250, 2018.

Zimmermann, V., and Gerber, N., “The Password Is Dead, Long Live the Password–A Laboratory Study on User Perceptions of Authentication Schemes”, International Journal of Human-Computer Studies, vol. 133, no. 1, pp. 26-44, 2020.

Spafford, E. H., “Observations on Reusable Password Choices”, in Purdue Technical Report, no. 92-049, pp. 1-14, 1992.

Wang, C., Jan, S. T., Hu, H., Bossart, D., and Wang, G., “The Next Domino to Fall: Empirical Analysis of User Passwords Across Online Services”, in Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy (CODASPY ’18), USA, pp. 196-203, 2018.

Pearman, S., Zhang, S. A., Bauer, L., Christin, N., and Cranor, L. F, “Why People (Don't) Use Password Managers Effectively”, in Proceedings of the Fifteenth Symposium on Usable Privacy and Security, Canada, pp. 319-338, 2019.

Alomari, R., and Thorpe, J., “On Password Behaviours and Attitudes in Different Populations”, Journal of Information Security and Applications, vol. 45, pp. 79-89, 2019.

O'Rourke, M., “The Year's Worst Password Offenders”, Risk Management, vol. 66, no. 1, pp. 36-36, 2019.

Dashline, “It’s the Most Unsecure Time of the Year: Worst Password Offenders 2021”, December 14, 2021.

Elizabeth Stobert and Robert Biddle, “The Password Life Cycle”, ACM Transactions on Security and Privacy, vol. 21, no. 3, Article 13, pp. 1-32, 2018.

He, D., Zhou, B., Yu, H., Cheng, Y., Chan, S., Zhang, M., and Guizani, N., “Group-Based Password Characteristics Analysis”, IEEE Network, vol. 35, no. 1, pp. 311-317, 2020.

Taha, M. S., Rahim, M. S. M., Lafta, S. A., Hashim, M. M., and Alzuabidi, H. M, “Combination of Steganography and Cryptography: A Short Survey”, in Proceeding of IOP Conference Series: Materials Science And Engineering, vol. 518, no. 5, pp. 1-14, 2019.

Deepthi, D. V. V., Benny, B. H., and Sreenu, K., “Various Ciphers in Classical Cryptography”, Journal of Physics, vol. 1228, no. 1, pp. 1-7, 2019.

Aung, T. M., Naing, H. H., and Hla, N. N., “A Complex Transformation of Monoalphabetic Cipher to Polyalphabetic Cipher: (Vigenère-Affine Cipher)”, International Journal of Machine Learning and Computing, vol. 9, no. 3, pp. 296-303, 2019.

Rachmawati, D., Hardi, S. M., and Pasaribu, R. P., “Combination of Columnar Transposition Cipher Caesar Cipher and Lempel Ziv Welch Algorithm in Image Security and Compression”, Journal of Physics, vol. 1339, no. 1, pp. 1-7, 2019.

Rihartanto, R., Supriadi, S., and Utomo, D. S. B., “Image Tiling Using Columnar Transposition”, in Proceeding of International Conference on Applied Information Technology and Innovation (ICAITI), Indonesia, pp. 118-123, 2018.

Kikani, R. J., Verma, K., Navalakhe, R., Shrivastava, G., and Shrivastava, V., “Cryptography: Recent Research Trends of Encrypting Mathematics”, Materials Today: Proceedings, vol. 56, no. 6, pp. 3247-7853, 2022.

Jagetiya, A., and Krishna, C. R., “Evolution of Information Security Algorithms”, In Design and Analysis of Security Protocol for Communication, pp. 29-77, 2020.

Santos, A., and Júnior, R. V., “Improving Caesar Cipher for Greater Security”, Kriativ-Tech, vol. 1, no. 9, pp. 1-7, 2021.

Singh, K., Johari, R., Singh, K., and Tyagi, H., “Mercurial Cipher: A New Cipher Technique and Comparative Analysis with Classical Cipher Techniques”, in Proceedings of IEEE International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), India, pp. 223-228, 2019.

Fadlan, M., Haryansyah, and Rosmini, “Three Layer Encryption Protocol: An Approach of Super Encryption Algorithm”, in Proceedings of the 3rd IEEE International Conference on Cybernetics and Intelligent System (ICORIS), pp. 1-5, 2021.

Dooley, J. F., “The Machines Take Over: Computer Cryptography”, In History of Cryptography and Cryptanalysis, Springer, pp. 167-184, 2018.

Zodpe, H., and Shaikh, A., “A Survey on Various Cryptanalytic Attacks on the AES Algorithm”, International Journal of Next-Generation Computing, vol. 12, no. 2, pp. 115-123, 2021.

Nyarko-Boateng, O., Asante, M., and Nti, I. K., “Implementation of Advanced Encryption Standard Algorithm With Key Length of 256 Bits for Preventing Data Loss in An Organization”, International Journal of Science and Engineering Applications, vol. 6, no. 3, pp. 88-94, 2017.