# AN INNOVATIVE PATTERN BASED PASSWORD METHOD USING TIME VARIABLE WITH ARITHMETIC OPERATIONS

Viral Panchal, RavirajPrajapati, Kalyani Patel

M.Sc. (CA & IT), K.S.School of Business Management, Gujarat University
Ahmedabad, Gujarat, India

## *ABSTRACT*

*In today's world, securing the assets is necessary that can be done by password. But imagine if password is stolen or hacked then what about the security of assets? In this Paper, we have discussed the major attacks as well as password authentication / security methods and techniques. We have proposed a password security method, where arithmetic operations are performed on user selected pattern from time variables to generate secure password. The task of validating the password or authentication of user can be done on both client and server side. We have analysed how proposed scheme defends across brute force, dictionary, phishing, shoulder surfing, key logger, video recording and replay attacks. To the best of our knowledge, our pattern based time variable password method with arithmetic operation is the one which is able to defend against the all major attacks together.*

## *KEYWORDS*

*Pattern Based Password Method, Brute Force, Dictionary, Key Logger, Shoulder Surfing, Phishing, Video Recording, Replay, Time Variable.*

## 1. INTRODUCTION

Password is used for securing the assets. Passwords play an important role in daily life in various computing applications like ATM machines, internet services, windows login, authentication in mobiles etc.[5] The major purpose of using password is to restrict unauthorised user access to the system. Most of the systems today rely on static passwords to verify the user's identity. Password may be in text, graphics or biometric. Users tend to use easy-to-guess passwords, use the same password in multiple accounts, write the passwords or store them on their machines etc. Furthermore, hackers have the option of using many techniques to steal passwords such as shoulder surfing, guessing, phishing etc. There are also well known password attacks namely brute force, dictionary, shoulder surfing, key logger, phishing, video recording, replay [2,5,8] etc. Hackers use these most common attacks to hack the password of any system like server's password. This all attacks have their own characteristics to break the password. However, some of them take some time and some of them steal the password easily in few minutes of time.

However, there are list of authentication technique available to defend the attacks namely conventional password scheme[2,5,8], Click Patterns[2,5,8], Keystroke Dynamics[2,5,8], Graphical Passwords [2,5,8], Biometrics [25,8], Authentication Panel [2,5,8], Reformation Based Authentication [5,8], Moving Balls Based [5,8], Security Scheme [5,8], Expression Based Security Scheme[5,8], Virtual Password [5,8], Time Signature [5,8] etc. These all techniques have their own advantages and disadvantages. While studying these methods we

DOI : 10.5121/ijist.2016.6215

found that any one technique is not applicable for all attacks. So we have proposed new method for securing password based on some patterns selected by users and password will be generated by the system. This method includes time variables like date, day, month, hour, minute, seconds for generating secure password. The pattern which is combination of time variable and arithmetic operation which is selected by the user based on their knowledge, requirement of confidentiality and their memory power. The method is discussed in detail in section IV of this paper.

## 2. TYPES OF ATTACKS

*Brute force attacks:* Brute force attacks are very time consuming as searching a password from all possibilities is a time taking process.For example, a user enters a password of 8 characters and all characters are lower case letters, then to break the password using the brute force attack it requires (26) combinations which is equal to 208827064576. If a single computer takes 1000 passwords to check in one second then total time will be 208827064576 / 1000 = 208827064.576 seconds which is equal to 58007.52 hours. This shows that brute force attack is effective for smaller passwords. [5, 8]

*Dictionary Attack:* The attacker makes the dictionary of most commonly used words that might have been be used as a password. The attacker then applies all these words to break the password. The dictionary attack is faster than brute force attack, but it has some limitations too i.e. Brute force attack contains limited words and sometimes it is unable to crack the password because there remains a possibility that password to be cracked may not be present in the dictionary itself. Many users generally write passwords related to the names of birds, familiar places, famous actors names etc. These passwords can be judged by the dictionary attack. [5, 8]

*Key Logger Attack:* The key loggers are the software programs which monitors the user activities by recording each and every key pressed by the user. The attacker installs the key logger software in the user system, either by installing that software himself or by tricking out the user to click to install that file in his (user) system. The key logger makes the log file of the keys pressed by the user and then sends that log file to the attacker's e-mail address. The attacker then gets the password and can access to the target system. [5, 8]

*Phishing Attack:* It is a web based attack in which the attacker redirects the user to the fake website to get passwords/ Pin Codes of the user. For example, suppose user wants to open website say "www.yahoo.com". The attacker redirects the user to another website e.g. "www.yah0o.com" whose interface is similar to that of the original website to disguise the user. The user then enters the login information which is retrieved by the attacker. The attacker then redirects the user to the original website and logins the user with the original website. [5, 8]

*Shoulder Surfing:* Shoulder Surfing is an alternative name of "spying" in which the attacker spies the user's movements to get his/her password. In this type of attack, the attacker observes the user; how he enters the password i.e. what keys of keyboard the user has pressed. [5, 8] There are many variations of shoulder surfing. The attacker can use binoculars to see the user entering the password from a distance. The attacker can listen that how many keys the user has pressed and then the attacker uses all the possibilities related to the password length to break it. [5]

*Video Recording Attack:* In such type of attack the attackers with the help of camera equipped mobile phone or miniature camera, analyses the recorded video of users which enters password. [5, 8] The attacker can use the hidden close circuit TV camera to observe the password.

*Replay Attack:* It is a way to attack challenge response user authentication mechanism. The method for this type of attack is that the attacker first enters his/her name in first login connection. To authenticate the user, the receiving device sends the challenge to the sender. [5]

## 3. RELATED WORK

Research and surveys on various attacks on password or vulnerability on authentication methods are described below with their source.

There are various password attacks and authentication techniques for user awareness [5] and also have countermeasure for each attacks [2]. Hence there is a Two Factor Authentication Method based on textual password as well as grid based password [8] that helps to provide security to authentication process. There are also lot of protocols used in authentication process that may have vulnerability, thus [6] providing protocols for login process that can keep track of online dictionary attacks only. The virtual password concept and adoption of user-determined randomized nonstop generation function is used for securing password attacks like shoulder surfing, key logger, phishing only [3]. However there is a method called 'One Time Password' (OTP) which also provides security to authentication process against some of attacks, but it needs more parties to be involved at the same time as well as it isbit complex and also have vulnerability while doing mobile activation process [4]. An innovative two factor authentication method called QRLogin which is limited to smart phone users only. The system combines the traditional username and password with time-sensitive and One Time Password. In this method user must have to convey to login by scanning a code with smart phone [7]. There is another method named icon based authentication method thatis simple and sufficiently secure evenwhen the authentication sequence being watched is proposed, but it is implemented on mobile data terminal that provides security against shoulder surfing attacks only [9].Password must be stored and transferred over network in encrypted form that can be done by most widely used method named MD5.

This MD5 hash function may provide security to password from theft but not surety on dictionary attack while it also used MD5 hash function to encrypt most common used password stored in its dictionary[1]. Here we addressed these major types of attacks and provide method to overcome from all problems of authentication methods and password attacks.However, our concept is compatible with existing Key Derivation Function, Password Based Key Derivation Function and Geo-Encryption (pass-phrase with real time GPS coordinates) [10]. It compensates the benefits provided by KDF, PBKDF and Geo-PBKDF without consuming CPU cycles on iteration for encryption and decryption to derive key.

## 4. OUR APPROCH

However, authentication/password security techniques are available for only a particular scenario. Hence we represent our solution as follows:

Every system needs protection and for that the authentication is needed to be given to authorized people to give right to access the system. The process of Authentication is consists of three parts:

1. System
2. User_id (or User)
3. Password

To authenticate the user, System needs to verify User_id via the user's Password which the user provides. In this procedure, system authenticates user by using User_id and Password.

All of System, User_id, and Password are fixed. It is very reasonable that a password should be constant for the purpose of easily remembering it. However, the price for easy to remember is, that the password can be stolen by others and then used to access the victim's account.

At the same time, we cannot put Password in a randomly variant form, which will make it impossible for a user to remember the password. To confront such a challenge, we propose a scheme using a new concept of pattern based password using time variable.

A pattern based password is a password which cannot be applied directly but instead generates a dynamic password which is submitted to the server for authentication. A pattern based password is composed of two parts,

1. Fixed Alphanumeric Password: - We denote it by "FA".
2. Time Variable Password: - We denote it by "TV".

Time Variable is different types of values of system-time which is frequently changed as per time.

1. Day:-We denote it by "d".
2. Month:-We denote it by "m".
3. Year:-We denote it by "y".
4. Hour:-We denote it by "h".
5. Minute:-We denote it by "M".
6. Weak day:-We denote it by "w".

Time variable can be discrete using operation (OP) which is listed below.

1. Plus(+)
2. Minus(-)
3. Multiplication(*)
4. Modular(%)

Time variable can be discrete using operation (OP) which is listed below. Now we have,

Password= FA + {TV} where,

FA = {0…9, a…z, A…Z, special characters}
TV = OP(t) where t = {d, m, y, h, M, w}

Here, TV is user specified operation on time-variable. Thus,

Password= FA +{TV1, …, TV24}

Here, 24 comes form 6*4 (6*4=24) where 6=no of TV, 4=no of OP.

The Most important thing about this method is that user can set the sequence of TV and FA as per his/her Choice like,

FA+TV1+…+TV24
TV1+TV2+…+FA+…+TV24
TV1+…TV24+FA

The user input includes (User_id, Password), where Password is pattern based password. On the server side, the server can also calculate Password in the same way to compare it with the submitted password.

Here, the password which is automatically generated by system which is a combination of time variable listed above and original password which is fixed alphanumeric. In this method user can set the sequence of this combination by providing priority of original password as well as pattern he/she has selected.

*System Flow:*

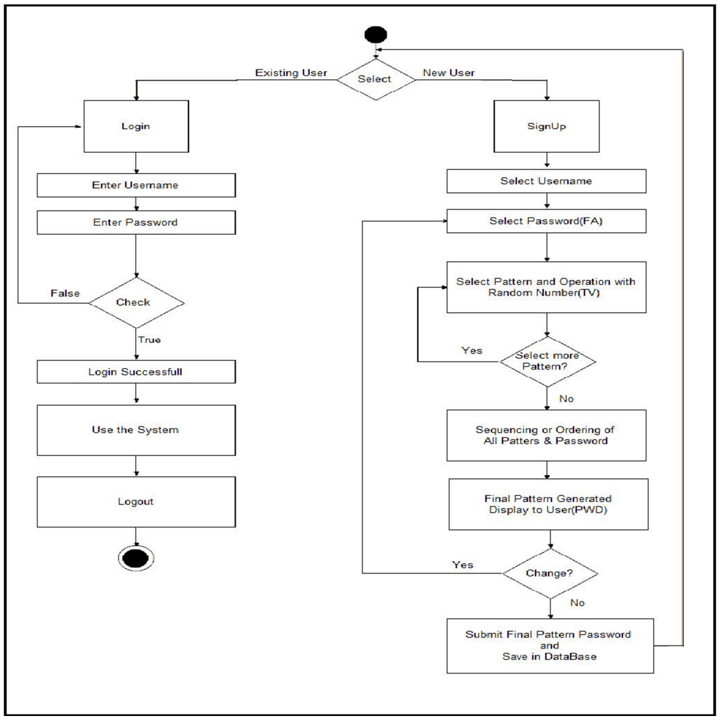
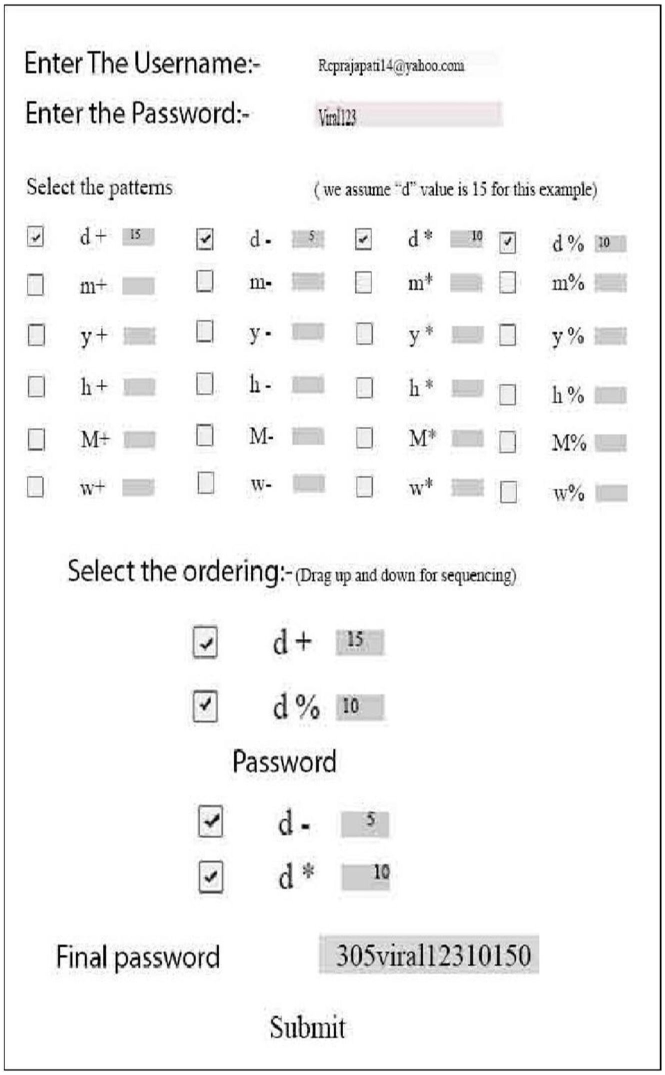

Existing User
Select
New User
Login
SignUp
Enter Username
Select Username
Enter Password
Select Password(FA)
False
Check
Select Pattern and Operation with Random Number(TV)
True
Yes
Select more Pattern?
Login Successfull
No
Use the System
Sequencing or Ordering of All Patters & Password
Logout
Final Pattern Generated Display to User(PWD)
Yes
Change?
No
Submit Final Pattern Password and Save in DataBase

Figure 1. Flow Chart of Suggested Solution

*Example of the method:*



Enter The Username:- Rcprajapati14@yahoo.com

Enter the Password:- Viral123

Select the patterns ( we assume "d" value is 15 for this example)

| | | | |
|---|---|---|---|
| ☑ d + 15 | ☑ d - 5 | ☑ d * 10 | ☑ d % 10 |
| □ m+ | □ m- | □ m* | □ m% |
| □ y + | □ y - | □ y * | □ y % |
| □ h + | □ h - | □ h * | □ h % |
| □ M+ | □ M- | □ M* | □ M% |
| □ w+ | □ w- | □ w* | □ w% |

Select the ordering:- (Drag up and down for sequencing)

☑ d + 15

☑ d % 10

Password

☑ d - 5

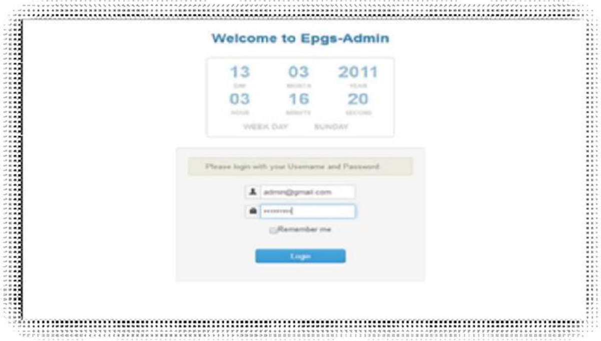
☑ d * 10

Final password 305viral12310150

Figure 2. Example of our system

In figure 2, the user’s original password is ‘viral123’ and let pattern is‘d’. Here, d is current date and for example value of‘d’ is 15. Select the arithmetic operation ‘+’, ‘%’, ‘-‘and ‘*’ on the assumed pattern‘d’.

After selecting pattern and arithmetic operations, user can set the priority of pattern and original password by dragging up and down. Finally, the password will be generated as shown in figure 2.



Welcome to Epgs-Admin

13 DAY 03 MONTH 2011 YEAR
03 HOUR 16 MINUTE 20 SECOND
WEEK DAY SUNDAY

Please login with your Username and Password

admin@gmail.com

□Remember me

Figure 3. Login Page

Figure 3 illustrate stimulating time synchronization for login in our system.
*Algorithm for authentication process*

```
Step: 1 - Enter Pattern
            Pattern is stored in USER_PAttern;
Step: 2 - Fetch Pattern from the database & store it in DB_PATTERN;
Step: 3 - Tokenizing the DB_PATTERN and stored it into the ARRAY_TOKEN. Step: 4
- Storing the size of array_token into variable N.
            N=sizeof (ARRAY_TOKEN);
Step: 5 -
            For (i=0; i<=N;i++)
            {
            If (ARRAY_TOKEN[i] != FA) /* e.g. FA = “abc123” */
            {DIGIT=Fetching Digit from Array_token[i]; /* e.g. from d + 10, it will fatch 10 */ /* Time
Variable */
            If (First Character of ARRAY_TOKEN [i] ==”d”)
            {TV=value of Current DAY;}
            Else If (First Character of ARRAY_TOKEN [i] ==”m”)
            {TV=value of Current MONTH;}
            Else If (First Character of ARRAY_TOKEN [i] ==”y”) {TV=value of Current YEAR;}

            Else If (First Character of ARRAY_TOKEN [i] ==”h”){ TV=value of Current HOUR;} Else If
            (First Character of ARRAY_TOKEN [i] ==”M”){TV=value of Current
            MINUTE;}
            Else If (First Character of ARRAY_TOKEN [i] ==”w”){TV=value of Current
            WEEKDAY; }
            /*Operation*/
            If (Second character of ARRAY_TOKEN [i]=”+”)
            {TV=TV+DIGIT; }
            Else If(Second character of ARRAY_TOKEN [i]=”-”){TV=TV-DIGIT;} Else
            If(Second character of ARRAY_TOKEN =”*”)
            {TV=TV*DIGIT;            }
            Else If(Second character of ARRAY_TOKEN =”%”){ TV=TV%DIGIT;} }/*end
if FA*/
Else    /*Pattern */
{
            TV=TV + FA;
 /* “+” sign is used for Concatenation  */
}
            FINAL_PAttern = FINAL_PAttern + TV; /*
“+” sign is used for Concatenation */
            }/*end for loop */
            If (USER_PAttern== FINAL_PAttern) {“
            SUCCESSFUL”}
            Else
            {“ FAILED”}
```

**Let us see tracing of the password authentication algorithm by taking “Date:15-8-2015, Time:12:12:12(HH:MM:SS) , Saturday**

Step:-1

USERNAME=viral@gmail.com

Step:-2

USER_PASSWORD=22viral@123120

Step:-3

DB_PASSWORD=viral@123

Step:-4

DB_PATTERN=M+10,Password,M*10

Step:-5

ARRAY_TOKEN[]={{M+10},{Password},{M*10}};

Step:-6

N=sizeof(ARRAY_TOKEN);
N=3

Step:-7

```
For(i=0;i<3;i++)
{
/* First Iteration when i=0(Processing First Token Which Is Pattern)*/
If(ARRAY_TOKEN[0]!=”Password”) /* Condition Become true Because
M+10!=”Password” */
{DIGIT=10 /*Extracting digit from pattern M+10 */
If(M==’d’)     /* Processing First Character Of Pattern M+10 */{/*Condition False!!*/}
If(M==’m’)     /* Processing First Character Of Pattern M+10 */{/*Condition False!!*/}
If(M==’y’)     /* Processing First Character Of Pattern M+10 */{/*Condition False!!*/}
If(M==’H’)     /* Processing First Character Of Pattern M+10 */{/*Condition False!!*/}
If(M==’M’)     /* Processing First Character Of Pattern M+10 */{/*Condition True*/
TV=12;            /* Current value of Minute */
}If(+==’+’)     /* Processing second Character Of Pattern M+10 */
{/*Condition True!!*/
TV = TV + DIGIT
TV = 12 + 10; TV =
22;
}
FINAL_PASSWORD = FINAL_PASSWORD + TV;
FINAL_PASSWORD = NULL + 22;
FINAL_PASSWORD = 22;
/* Second Iteration when i=1(Processing second token which is Password)*/
if(ARRAY_TOKEN[1]!="Password")
{/*Condition False BecauseARRAY_TOKEN[1]="Password" */ }
Else{TV = TV + DB_PASSWORD;
        TV = viral@123}
        FINAL_PASSWORD = FINAL_PASSWORD + TV;
        FINAL_PASSWORD = 22 + viral@123
        FINAL_PASSWORD = 22viral@123
        /* Third Iteration when i=2(Processing Third token which is Pattern)*/
        If(ARRAY_TOKEN[2]!="Password") /* Condition Become true Because
        M*10!="Password" */
        {DIGIT=10    /*Extracting digit from pattern M*10 */
        If(M=='d')          /* Processing First Character Of Pattern M*10 */
        {/*Condition False!!*/}
        If(M=='m')      /* Processing First Character Of Pattern M*10 */
        {/*Condition False!!*/}
        If(M=='y')          /* Processing First Character Of Pattern M*10 */
        {/*Condition False!!*/}
        If(M=='H')      /* Processing First Character Of
        Pattern M*10 */{              /*Condition False!!*/}
        If(M=='M')      /* Processing First Character Of
        Pattern M*10 */
        {/*Condition True*/
        TV=12;              /* Current value of Minute */
        }
        If(*=='+')      /* Processing second Character Of Pattern M*10 */
        {/*Condition FAlLSE!!*/}
        If(*=='-')          /* Processing second Character Of Pattern M*10 */
        {/*Condition FAlLSE!!*/}
        If(*=='*')          /* Processing second Character Of Pattern M*10 */
        {
        /*Condition TRUE*/
        TV = TV * DIGIT;
        TV = 12 * 10;
        TV = 120;}
FINAL_PASSWORD = FINAL_PASSWORD + TV;
FINAL_PASSWORD =22viral@123 +120;
FINAL_PASSWORD = 22viral@123120;
        }
}   /* For Loop Ended. */
If(USER_PASSWORD==FINAL_PASSWORD)  /* 22viral@123120==22viral@123120 */
{/* Condition True… */
        LOGIN SUCCESSFUL…
}
```

## 5. STUDY OF ATTACKS ON OUR APPROCH

*Our solution provides the security by defending several attacks discussed below.*

*Brute force Attack:*The major disadvantage of Brute force Attack is it take Too much time.[5,8] In our method password is frequently changing and it is being near too impossible for brute force attack to attempt all the combination before the password is changed.

*Dictionary Attack:* The major disadvantage of Dictionary attack is it attempts only to a targeted list of weak passwords or attempts on a limited number of key combinations that has a high possibility of getting succeeded.[2] In our method the password is combination of pattern and original password which is not common or simple. Thus it is not in the list of dictionary of the attack.

*Phishing Attack:* In Phishing attack, Attacker gets the password through sniffing or eavesdropper while password is transmitted over the network. But in our solution password is being frequently updated so whatever the password is stolen by attacker it becomes useless for him.

*Keylogger Attack:* Key logger store each and every keystroke of the system in which password is also included. That password can be used by attacker to attack to the system. But in our solution password is being frequently updated so attacker couldn't use that password for getting the access to the system.

*Shoulder Surfing Attack:* In our solution Password is changed frequently so Shoulder surfing attack can't happen because the password which attacker gets through Shoulder Surfing is outdated.

*Video Recording Attack:* Video Recording attack is making the video of user while he is entering the password. But video also can't get the right password because whenever attacker tries to fetch the password from video it is already outdated.

*Replay Attack:* Replay attack is also can't be successful against our solution although the password in it is in simple text and password is being changed frequently.

*Comparative Analysis*

The table 1, shows the comparative analysis of different authentication techniques/methods which helps to protect password from attacks[5]. Our method does not require any hardware as it has resistance from all major attacks while here in this table not any method/technique that are applicable to defend all major password attacks. However our method is for security used by smart person. It gives high security with less effort.

Table 1. Comparative analysis of authentication methods

| **Method** | **Resistance To Attacks** | **Additional Hardware Require-Ment** | **Cost** | **Mental Attitude Effects** | **Protection Level** | **Processing Time** |
|---|---|---|---|---|---|---|
| Conventional Password Scheme | No | | Normal | Yes | Low | Fast |
| Key Stroke Dynamics | Shoulder Surfing, Phishing, Key Loggers | No | Normal | Yes | Medium | Medium |
| Click Patterns | Shoulder Surfing, Phishing, Key Loggers | No | Normal | Yes | Medium | Medium |
| Graphical Passwords | Shoulder Surfing | Yes | | Yes | Medium | Slow |
| Biometrics | Shoulder Surfing, Phishing, Key Loggers Etc | Yes | | No | High | Slow |
| Authentication Panel | Video Recording, Shouldering | No | Normal | Yes | High | Medium |
| Reformation Based | Brute Force, Video Recording, Shouldering And Dictionary Attacks | No | Normal | No | Medium | Fast |
| Moving Balls Based | Dictionary Attacks, Shouldering | No | Normal | Yes | High | Medium |
| Expression Based | Brute Force, Video Recording, Shouldering And Dictionary Attacks | No | Normal | Yes | High | Fast |
| Virtual Passwords | Phishing, Key Loggers And All Other Online Attacks | May | May Be High | No | Medium | Fast |
| Time Signature | Shoulder Surfing, Dictionary Attacks, Replay Attacks, Key Loggers Etc | No | Normal | Yes | High | Slow |

## 6. CONCLUSIONS

The paper discussed the challenges and major attacks on password and the common techniques which are used to protect the password. However one method or technique does not provide the security from the majority of attacks. We have proposed the pattern based time variable password scheme that includes some arithmetic operation on pattern. This method is totally user friendly and the user has option to choose the pattern and arithmetic operation for generating secured password. To the best of our knowledge, this method provides the security to the user's password against majority attacks. However, simplicity and security both are necessary but is difficult to achieve both.

We proposed one of the best schemes to protect password against different attacks. But in case of social engineering our system may fail because the user itself being fool and give the password pattern to attacker to knowingly or unknowingly. In case when user is forgets the pattern of password, then we will propose some other authentication method for user to check his identity and give him access to the system. In such kind of situation whatever method is used it's dis-advantages is also our systems disadvantage. Eg: we can use OTP (One Time Password) to check user identity but OTP has dis-advantage that if the device is in the wrong hand in which OTP is send then the security purpose is not being fulfilled.

## REFERENCES

[1] JayeetaMajumder. "Dictionary Attack on MD5 Hash." International Journal of Engineering Research and Applications, vol. 2, no. 3, pp. 721-724, May-June 2012.

[2] Jesudoss A, Subramaniam N.P. "A Survey On Authentication Attacks And Countermeasures In A Distributed Environment." Indian Journal of Computer Science and Engineering (IJCSE), vol. 5, no. 2, pp. 71-77, April-May 2014.

[3] Kanagaraj S, JavithIbramSha M, MadhanKumaran D, Rajkumar D. "Differentiated Virtual Passwords For Protecting Users From Password Theft." International Journal of Engineering Reserch and Science & Technology, vol. 2, no. 2, pp. 93-100, May 2013.

[4] M. VijuPrakash, P. Alwin Infant and S. JeyaShobana. "Eliminating Vulnerable Attacks Using One Time Password and PassText – Analytical Study of Blended Schema." Universal Journal of Computer Science and Engineering Technology vol. 1, no. 2, pp. 133-140, Nov 2010.

[5] MudassarRaza, Muhammad Iqbal, Muhammad Sharif, WaqasHaider. "A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication." World Applied Sciences Journal, vol. 19, no. 4, pp. 439-444, 2012.

[6] Shabana T Pirjade, P. K. Deshmukh. "Defend Against Online Password Guessing Attacks."International Journal of Advanced Research in Computer Sci ence and Software Engineering, vol.4, no. 9, pp. 106-111, September 2014.

[7] SoonduckYoo, Seung-jang Shin, Dae-hyunRyu,. "An Innovative Two Factor Authentication Method TheQRLogin System." International Journal Of Secure And Its Application, vol. 7, no. 3, pp. 293-302, May 2013.

[8] SunainaOberoi, Harmandeep Singh "Design & Development of Two Factor Hash Based Authentication Framework." International Journal of Computer Science and Information Technology Research, vol. 2, no. 4, pp. 199-203, October - December 2014.

[9] Yoshihiro Kita, Fumio Sugai, MiRang Park, Naonobu Okazaki. "Proposal and its Evaluation of a Shoulder-Surfing Attack Resistant Authentication Method:Secret Tap with Double Shift."International Journal of Cyber-Security and Digital Forensics, vol. 2, no. 1, pp. 48-55, 2013.

[10] J.Zdziarski, Hacking and Securing IOS Applications : Stealing data, Hijacking Software & How to Prevent , 1st ed. , O'reilly,2012

**Authors**

**Kalyani A. Patel, (B.E., M.C.A. Ph.D.)**
Currently working as an Assistant Professor M.Sc.(CA & IT), K.S. School of business Management, Gujarat University . Her research area is: Software Engineering, Information Security, Big Data Analytics, NLP , Information Retrieval,

Viral Panchal
B.Sc(CA & IT) GPA:3.05
M.Sc(CA & IT) pursuing
M: 9409602632

Raviraj Prajapati
B.Sc(CA & IT) GPA:3.44
M.Sc(CA & IT) pursuing
M: 8672015134