Enhanced Authentication Schemes for Intrusion Prevention using Native Language Passwords

Sreelatha Malempati [1], Shashi Mogalla [2]

[1]Dept. of Computer Science & Engineering,
R.V.R. & J.C. College of Engineering, Chowdavaram, Guntur, A.P

[2]Dept. of Computer Science & System Engineering,
Andhra University College of Engineering, Andhra University, Visakhapatnam, A.P.

Abstract: Textual password authentication schemes are vulnerable to attacks like eves dropping, shoulder surfing and hidden cameras. An authentication scheme which integrates graphics and text is resistant to these attacks. N ative language passwords are more secure than standard language passwords. Users can remember native language passwords better than any other language, by nature. A shape based textual password authentication scheme using native language of the user is discussed and enhancements are proposed to make the authentication scheme more secure.

Keywords: *shape and text based authentication, textual password, native language passwords, intrusion prevention*

1. Introduction: Authentication is the first phase of the security of any system. Textual password is the most popular method used for authentication. In any authentication scheme, the length of the password plays major role. The intruder should not be able to crack the password within the expiry time of passwords. The password should not be too short or too easy to guess which makes intruder's job easy. If a password is hard to guess, then it is difficult to remember. If the password is too long, user may write his passwords on paper which is prone to theft and social engineering. Therefore, human remembrance is very important for authentication. A good authentication scheme should be resistant to dictionary attack, shoulder surfing attack & hidden camera and it should be easy to use.

Graphical passwords provide an alternative solution to textual passwords, where a user selects a picture or something related to a set of visual representations. It is believed that humans recall a picture better than text. Graphical password schemes are considered as more secure and resistant to dictionary attacks than textual passwords. But, simple graphical schemes are vulnerable to shoulder surfing and hidden camera attacks. An alternative solution is to use biometrics for authentication using fingerprints, iris, face recognition and hand signature. Biometrics are susceptible to damage and physical changes which causes authentication scheme to fail. A good solution is to use an authentication scheme which integrates graphics and text. Native language passwords are more secure than standard language passwords because dictionary attack is not possible. Users can remember native language passwords better than any other language, by nature. In this paper, a s hape based textual password authentication scheme using native language of the user is discussed and enhancements are proposed to make the authentication scheme more secure. The native language of the authors of this paper "**Telugu**" is used for analysis.

This paper is organized as follows: Related work is discussed in section 2, in section 3 the shape based textual authentication scheme using native language of the user is discussed, enhancements are proposed in section 4, conclusion is given in section 5.

2. Related work

The alternatives to text-based schemes are graphical password schemes and many schemes have been proposed using images and figures. A graphical password scheme designed by Blonder[1] used an image. During r egistration user clicks on several locations on the image. During authentication, the user must click on the approximate areas of the locations. Dhamija and perrig [2] proposed a graphical authentication scheme in which the user selects a certain number of images from a set of random pictures. Later user has to identify the pre-selected images for authentication. Jansen [4,5] proposed a graphical password scheme for mobile devices. During password creation, a user selects a theme consisting of photos in thumbnail size and set a s equence of pictures as a password. During authentication, user must recognize the images in the correct order. Each thumb nail image is assigned a n umerical value, thus the sequence of the chosen images will create a numerical password. As the no. of images is limited to 30, the password space of this scheme is not large.

Three authentication schemes based on picture recognition, object recognition, & pseudo word recognition were designed by Weinshall and Kirkpatrick

[11] and conducted user studies of these schemes. The results declared that pictures are most effective than the other two proposed schemes. A graphical password authentication scheme “passdoodle” was designed by Goldberg [3]. In this scheme, handwritten design or text should be drawn with a stylus onto a touch sensitive screen. Jermyn et al [6] proposed a technique called “Draw A Secret”(DAS) where a user draws the password on a 2D grid. The coordinates of this drawing on the grid are stored in order. During authentication user must redraw the picture. The user is authenticated if the drawing touches the grid in the same order. All these graphical authentication schemes are vulnerable to shoulder surfing.

Most of the above schemes were vulnerable to shoulder surfing. To overcome this problem, many techniques were proposed. Zhao and Li [12] proposed a shoulder-surfing resistant scheme “S3PAS” . In this, during login stage, user must find his original text password in the login image and click inside the invisible triangle region. The system integrates both graphical and textual password schemes. Man, et al, [8] proposed another shoulder-surfing resistant technique. In this scheme, a user chooses many images as the pass-objects. The pass-objects have variants and each of them is assigned to a unique code. In the authentication stage, the user must type the unique codes of the pass-objects variants in the scenes provided by the system. Although the scheme shows perfect results in resisting hidden camera, it requires the user to remember code with the pass-object variants.

Luca et al. [7] proposed a stroke based shape password for ATMs. They argued that using shapes will allow more complex and more secure authentication with a lower cognition load. More graphical password schemes have been summarized in a recent survey paper [10]. Zheng et al [13] designed a grid- based approach as hybrid password scheme based on shape and text. The basic concept is mapping shape to text with strokes of the shape and a grid with text. The user is proposed to select a shape which can be a number, character, geometric shape or a random shape. Selecting simple and common shapes makes password cracking easy. User can not remember random shapes though they are strong. Remembering native language password is better than any other language or arbitrary shapes for the user. Sreelatha Malempati and Shashi Mogalla [9] proposed an authentication scheme based on native language passwords. This paper discusses the same technique and proposes enhancements to that technique.

3. Authentication scheme

The authentication scheme consists of three steps:

- password creation
- password entry
- password verification

3.1 Password Creation

User selects a character from his native language character set. Each character may contain one or more strokes. A stroke is an ordered list of cells. A password is represented by a sequence of strokes. The length of a stroke is the number of cells it contains. The length of the password is the sum of the lengths of its strokes. An interface consisting of a grid of size 5x 5 will be displayed on the screen. User has to select an ordered list of grid cells to represent the shape of the character selected for password. Consider the character in fig 1.

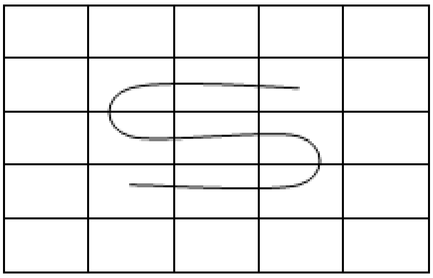


Fig 1: The password character

This character consists of two strokes each consisting of a set of ordered grid cells. The first stroke of the character (Fig 2) starts from the grid cell (2,4) and ends with (4,2). The second stroke (Fig 3) starts with (1,2) and ends with (1,4).

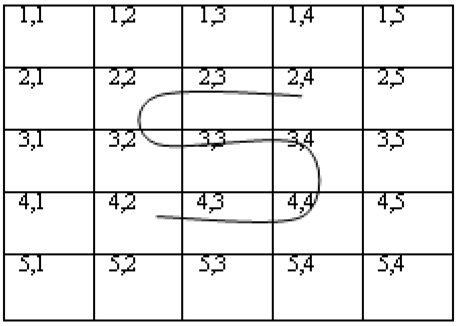


Fig 2: The first stroke of the character

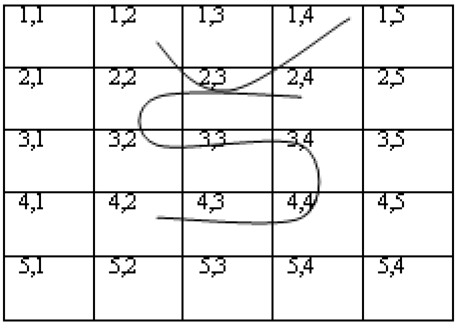


Fig 3: The two strokes of the character

Totally the shape of the character can be represented by the grid cells { (2,4), (2,3), (2,2), (3,2), (3,3), (3,4), (4,4), (4,3), (4,2), (1,2), (2,3), (1,4) }. User has to select the grid cells in this order at the time of password creation.

3.2 Password Entry

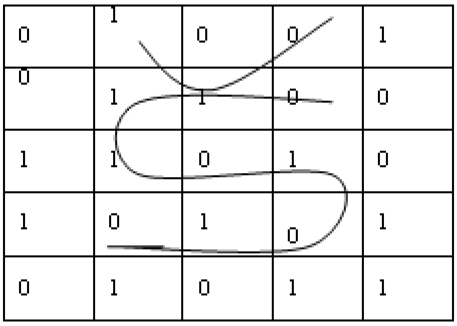
At the time of login, user has to enter his login name and password. An interface consisting of grid of size 5*5 will be displayed. The grid contains a symbol in each cell. Based on the symbols in the grid cells , the user has to enter the password. For the interface in fig 4, suppose the user enters the password {011101010110}.

3.3 Password Verification

After password entry, the authentication scheme will verify the password. It will compare the symbols of the interface in the positions of the grid cells selected by the user at the time of password creation with the elements of the password entered by the user. If the password entered is not correct, the system will generate another login interface grid with different symbols. At each login step, the symbols vary, but the shape of the character and the order of the grid cells will not vary. So, the text-based brute force attack will not work. For the above interface, the password will be verified in this manner : The shape of the character is represented by the cells : { (2,4), (2,3), (2,2), (3,2), (3,3), (3,4), (4,4), (4,3), (4,2), (1,2), (2,3), (1,4) }. For this interface, by considering the symbols of the cells in the above order, actual password is 011101010110 and the password entered by the user is 011101010110. In this example, the user is authenticated.

1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 4: Login interface grid with symbols



0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 5: grid with shape of the character

4. Enhanced schemes

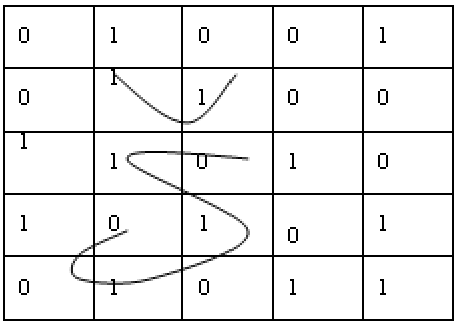
4.1 password entry

The authentication technique can provide many options to the user for inputting the password. The options provided are inputting octal digits, interchanging digits, adding redundant digits and inverting bits. The intruder has no information about the option selected by the user, so he has to try all the options in order to break the system.

(i) Octal digits

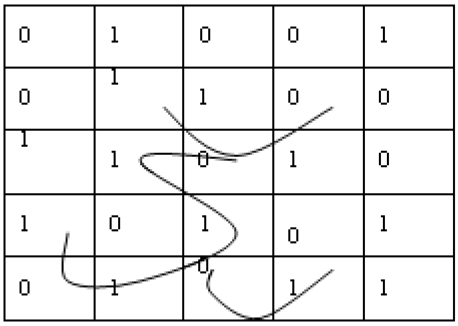
User has to enter a long password during login which is a vulnerable activity. Instead of entering binary data, user can enter octal digits.

For the login interface grid in fig:4 , user has to enter {011101010110}. Instead of this, user can enter 3526 {011 101 010 110 }as password which makes entry easy. If the password do not contain no of bits equal to an integral multiple of 3, then padding can be done at the end. Single bit padding may add either 1 or 0 and two bits padding may add 00/01/10/11 to confuse the intruder.



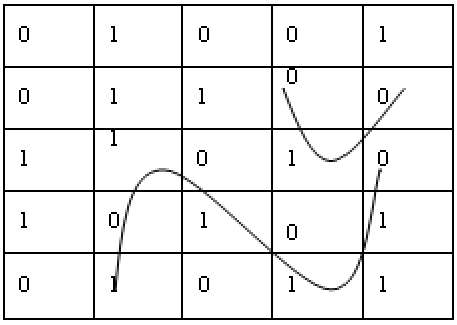
0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 6: the character “ka” with 10-bits



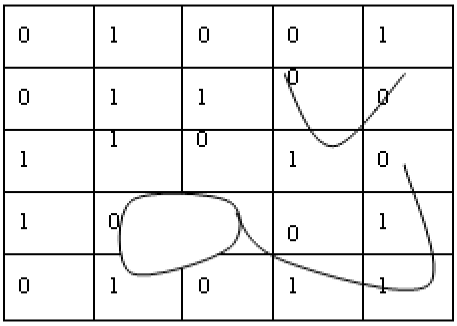
0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 7: the character “ku” with padding “00”



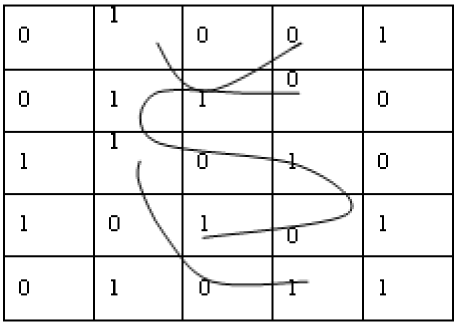
0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 8: the character “sa”



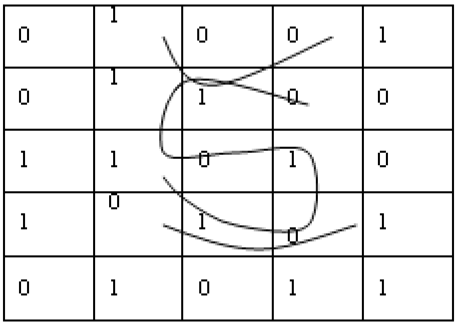
0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0		0	1
0	1	0	1	1

Fig 9: the character “pa”



0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

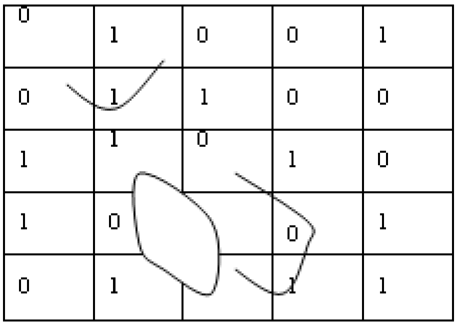
Fig 10: the character “kra” with “352***1***6”



0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

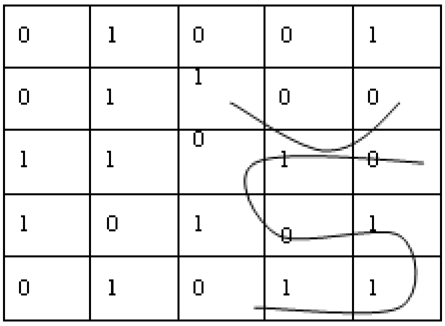
Fig 11: the character “kra” with “35***3***26”

(iv) **Inverting bits**: The user can invert all bits in the password or invert bits in the alternate octal digits to give wrong information to the intruder. For the input 3526, by inverting all bits in the password user can enter 4251 or by inverting alternate digits, he can enter 3221 or he can invert the middle two digits and make the password as 3256. The characters in fig:12 and fig:13 may appear on the grid.



0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0		0	1
0	1		1	1

Fig 12: the character “pa” with “3221”

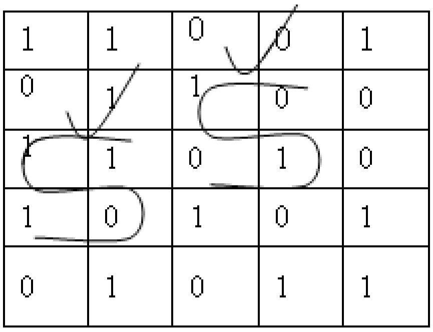


0	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 13: the character “ka” with 3256

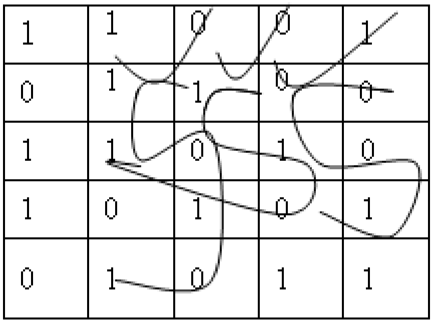
4.2 Variants of password

Suppose the attacker has no information about login interface grid, the length of the password and the password. Then he has to try all of the shapes and characters. When he has information that the user’s password is “ka”, he has to verify all variants of “ka” at different locations. T he character may be at different locations (fig:14) and may have many variants (fig:15).



1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 14: “ka” at different locations



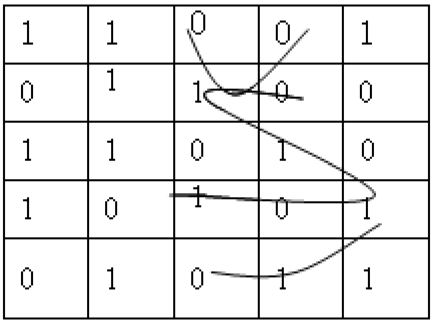
1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 15: variants of "ka"

4.3 Composite / Complex Characters

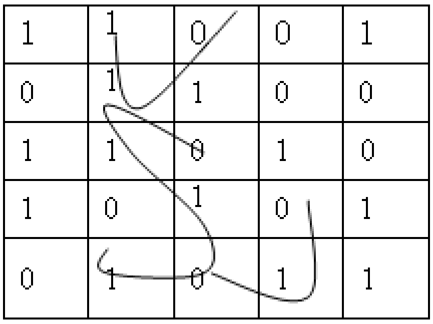
The character set contains composite or complex characters. Composite characters contain many simple strokes. The complex character contains closed and difficult strokes.

(i) Composite character: If t he user uses composite character, then it is difficult for the intruder to break the password even with the knowledge of login interface grid and the password. Consider the password {011101010110}.



1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 16: the character "kra"

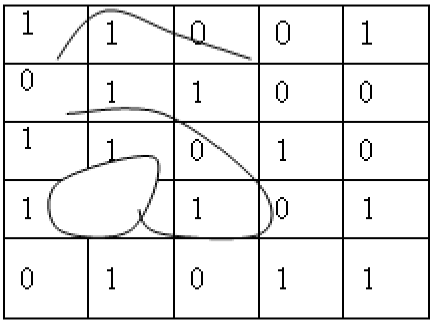


1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 17: the character "ku"

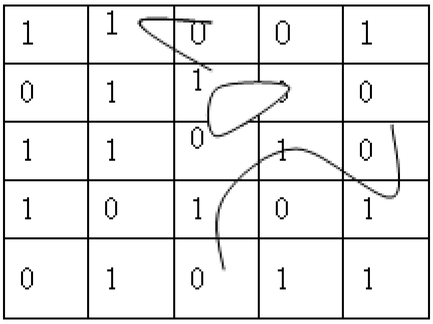
The letter "kra" (fig:16) consists of three strokes- first one is {011101}, second stroke is {110} and the third stroke is {110}. Generally the third stroke goes from left to right but user may be having the style of writing from right to left or he may select it to confuse the intruder. Other possibilities include the character "ku" (fig :17) .

(ii) **Complex character**: when the user selects complex character as password, it is very difficult for the intruder to detect the password. At the same time, it is a bit difficult job for the user to remember the locations of the grid cells. Two examples are given in fig:18 and fig 19.



1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1		1	0	1
0	1	0	1	1

Fig 18: the character "aee"



1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 19: the character "see"

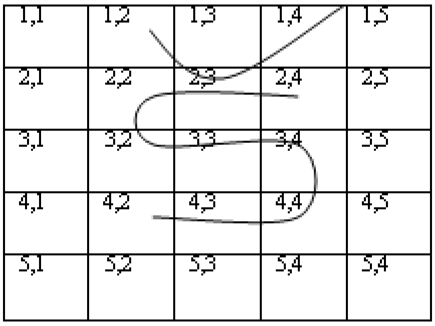
For composite and complex characters, users can select part of the character instead of using the entire character as password. User can definitely remember even part of the character than any other arbitrary shape.

4.4 Flexibility

Some users may be able to remember the shape of the character but not the rows or columns perfectly. In such a case, the user can inform the system to accept password with either horizontal drift or vertical drift. User can put a restriction on how many rows or columns drifting is allowed. If the grid size is large, then drifting may be possible for number of rows and columns.

(a) Horizontal drift

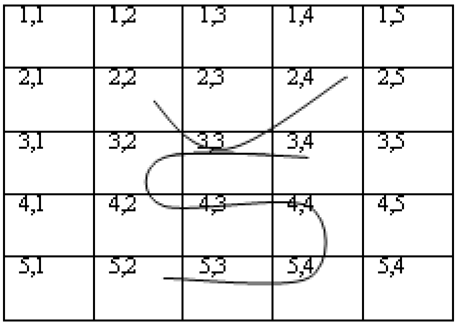
Columns remain same, but rows may change (up or down). Consider the character "ka" in fig:20. Here the order of the grid cells for the character is { (2,4), (2,3), (2,2), (3,2), (3,3), (3,4), (4,4), (4,3), (4,2), (1,2), (2,3), (1,4) }



1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,4

Fig 20: character “ka” (actual)

Consider the character “ka” after horizontal drift in fig:21. Here, the order of grid cells is { (3,4), (3,3), (3,2), (4,2), (4,3), (4,4), (5,4), (5,3), (5,2), (2,2),(3,3), (2,4) } .Every cell (x,y) is to be considered as (x+1,y).

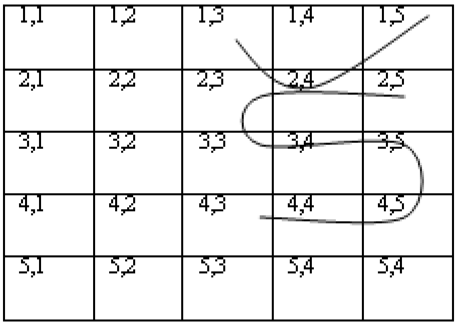


1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,4

Fig 21: the character “ka” (after horizontal drifting)

(b) Vertical drift

Rows remain same but columns may change (left or right). Consider the character “ka” after vertical drifting in fig:22 Here, the order of grid cells is { (2,5), (2,4), (2,3), (3,3), (3,4), (3,5), (4,5), (4,4), (4,3), (1,3), (2,4), (1,5) } Every cell (x,y) is to be considered as (x,y+1).

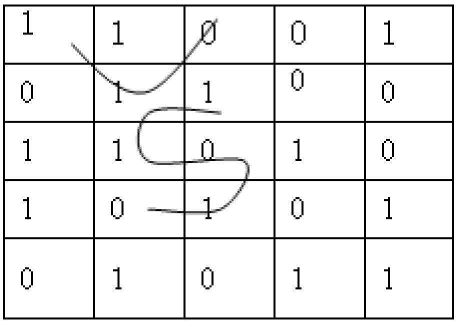


1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,4

Fig 22: the character “ka” (after vertical drifting)

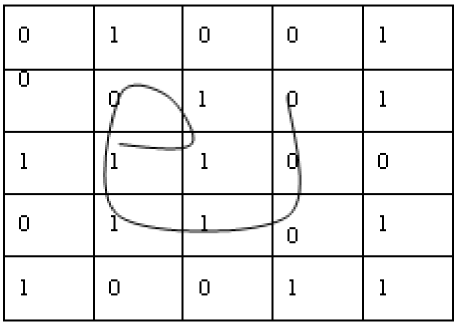
4.5 Extensibility

If the user requires additional security, he can add one more level to the authentication process by adding one more character. It is easy for the user to remember a two letter word of native language like “kala” (fig 23, fig 24) or “kadha”. When the user enters the first character, the authentication system validates it. If it is correct, then another login interface grid is displayed requesting the user to enter the second character. If the second character is also correct, then the user is authenticated.



1	1	0	0	1
0	1	1	0	0
1	1	0	1	0
1	0	1	0	1
0	1	0	1	1

Fig 23: the character “ka”



0	1	0	0	1
0	0	1	0	1
1	1	1	0	0
0	1	1	0	1
1	0	0	1	1

Fig 24: the character “la”

5. CONCLUSION

In this paper an authentication scheme based shape and text using native language passwords is discussed and some enhanced schemes of the basic technique are proposed. The native language of the authors of this paper “Telugu “ is considered for analysis. The proposed schemes are resistant to eves dropping, brute force attack, shoulder surfing and hidden camera. The usability and adaptability of the proposed schemes should be verified.

REFERENCES

[1] G. E. Blonder, “Graphical Passwords,” in Lucent Technologies, Inc., Murray Hill, NJ, U. S. Patent, Ed.United States, 1996.

[2] R. Dhamija and A Perrig, "Deja Vu: A User Study using Images For Authentication", 9th USENIX Security Symposium, 2000.

[3] J. Goldberg, J. Hagman, V. Sazawal, "Doodling Our Way To Better Authentication", CHI '02 extended abstracts on Human Factors in Computer Systems, 2002.

[4] W. Jansen, "Authenticating Mobile Device User through Image Selection," in *Data Security*, 2004.

[5] W. Jansen, "Authenticating Users on Handheld Devices “in *Proceedings of Canadian Information Technology Security Symposium*, 2003.

[6] Jermyn, I., Mayer A., Monrose, F., Reiter,M., and Rubin., “The design and analysis of graphical passwords” in proceedings of USENIX Security Symposium, August 1999.

[7] A. D. Luca, R. Weiss, and H. Hussmann, "PassShape:stroke based shape passwords," in *Proceedings of the conference of the computer-human interaction special interest group (CHISIG) of Australia on Computer-human interaction: design: activities, artifacts and environments*. 28-30 November 2007, Adelaide, Australia, pp. 239-240.

[8] S.Man, D. Hong, and M.Mathews, "A shouldersurfing resistant graphical password scheme," in *Proceedings of International conference on security and management. LasVergas*, NV, 2003

[9] Sreelatha Malempati and Shashi Mogalla, “Intrusion Prevention by Native Language Password Authentication Scheme” ,4th International conference on network security and its applications CNSA 2011, Springer LNCS-CCIS 196, pp. 239–248

[9] X. Suo, Y. Zhu, and G. S. Owen, "Graphical passwords: A survey," *21st Annual Computer Security Applications Conference (ASCSAC 2005)*. Tucson, 2005.

[10] D. Weinshall and S. Kirkpatrick, “Passwords You’ll Never Forget, but Can’t Recall,” in Proceedings of Conference on Hman Factors in Computing Systems (CHI), Vienna, Austria: ACM, 2004.

[11] H. Zhao and X. Li, "S3PAS: A Scalable Shoulder-Surfing Resistant Textual-Graphical Password Authentication Scheme," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW 07)*, vol. 2. Canada, 2007, pp. 467-472.

[12] Z. Zheng, X. Liu, L. Yin, Z. Liu “A Hybrid password authentication scheme based on shape and text” Journal of Computers, vol.5, no.5 May 2010