Free text, Keystrokes dynamics, Biometrics,
Digraph Grouping, Dense Area Identification

Jayasree SAHA[1], Rituparna CHAKI[1]

AN APPROACH TO CLASSIFY KEYSTROKE PATTERNS FOR REMOTE USER AUTHENTICATION

The authentication of users is of utmost importance in remote applications such as healthcare, banking, stock markets, etc. Key stroke dynamics are popular biometrics tools used for this purpose. Continuous authentication requires free text analysis which has a number of challenges. This paper has proposed a solution to identify the existence of a unique pattern in each individual user's keystroke dynamics. However, dense zone identification is important factor in forming the intelligent database of user profile for authentication. The authors have categorized basic key stroke features of digraph into 57 groups depending on distance traversed while moving from one key to another. The paper also includes graphical plots of the grouping of time vector which has unveiled some characteristics of overlapping typing style of users. The authors hope to extend this logic for identifying behavioral disorders in users.

1. INTRODUCTION

Society today is becoming highly dependent on pervasive computing for improved living conditions. These systems involve various private and public organizations, which need stringent protection systems for the data. This requirement necessitates the use of a combined security mechanism for data transaction as well as authentication of the user. Generally, one time user authentication is performed by password protection method. However, requirement of continuous authentication in any application entails study of free text key stroke analysis for ensuring the authenticity of a user.

Due to cost effectiveness behavioral biometrics is common choice for user authentication. They are only susceptible to environmental and psychological changes. In this paper, the authors have chosen keystroke dynamics for authentication which is a behavioral biometrics and involves collection of typing rhythm of individual users. The dynamics are based on key press time and key release time of each character being typed. Rest of the features are evaluated depending on this time stamp. Dwell time and flight time are two common features extracted from fixed text string for key stroke analysis. Fixed text string with dwell time (the time span during which a key is pressed) and flight time (time span from one key release time to next key press time) has been found to be useful in one time static authentication but it is not suitable to detect above features for all possible pair of keys. It is required to verify a user dynamically without interrupting his work while monitoring him continuously. This can be achieved either by having

[1] University of Calcutta, India

all possible pair of keys or adding some special features which may substitute all possible pair of keys. The major concern of key stroke authentication is in its stability as human behavior characteristics changes in short time and availability of all possible pair of keystrokes during enrollment period. The main challenge is to get all possible key strokes in a small sample size. This paper presents the logic for identifying the density of occurrence of specific groups of keystrokes in the typing pattern of user. The findings are compared logically to the logics presented in [1], [3]. The authors have restricted their experiment to English letter (A to Z) keywords in key stroke pattern analysis. They have selected flight time as key stroke feature and analyzed them for possible grouping, using distance factor which make them free to consider each letter separately.
The rest of the paper is structured as follows. In section 2, the state of the art study for fixed and free text analysis for key stroke authentication is presented. This section also includes the present research gap. Section 3 describes the behavior modeling difficulties in enrollment phase and present the proposed solution. Section 4 shows the result sets of the experiments. Section 5 concludes the paper.

2. STATE OF THE ART IN KEY STROKE AUTHENTICATION

The most basic form of keystrokes authentication involves fixed text analysis.This consists of collection of fixed texts during enrollment phase and then processed them considering two features dwell time and flight time [1], [2], [3], [5], [7], [8]. Depending on the results of the processing, user profile is created. During verification phase user signature is created by providing same phrase/word as in enrollment time [2]. Static authentication however lacks the modes to handle the complexities involved in pervasive computing. Non-static biometric technique like in [4] may be used to identify users, based on analyzing habitual rhythm patterns in the way they type. John A. Robinson et al. used minimum intra-class distance (MICD) classifier, nonlinear classifier, inductive learning classifier to classify typed login signature [8] and found inductive learning classifier as the best classifier which has imposter acceptance rate 10% and valid user rejection rate 9%. Neural Network [6], [10] and fuzzy logic [10] are found to be a good performer in fixed text analysis in password protection method.
With the advancement of technology more stringent security requirement opens the subject of continuous authentication (CA). There are several challenges in the field of continuous authentication. One challenge is to approximate all the key strokes that are not present in fixed text used in enrollment phase. It is important as all keys are not collected during enrollment phase. In [3], a method of free text analysis is proposed by classifying the keyboard keys in 8 parts. Terence Sim et. al. has investigated an issue on whether digraphs and tri graphs are just as discriminative for free text as they are for fixed text [9]. In [1], it is assumed that most frequently occurring keys (digraph/monograph) are covered during enrollment phase and rest keys are evaluated from the existing one using key mapping technique combined with neural network. Digraph key mapping table and monograph key mapping table usually represents user profile signature. They have put special emphasize on making digraphs signature and monograph signature.
The authors had collected data of 60 users and found FAR 0.0152% and FRR 4.82% when 77% threshold has been set to the decision maker module. They performed similar experiment with 53 users keeping threshold for decision making module same. It was found that when Decidability factor lies between 0.4 to 0.6, FAR tends to zero and FRR to 4.82% to 5.12%.

2.1. OBJECTIVE OF THE PROPOSED SOLUTION

It is ovserved from the above discussion that free text analysis gives more reliable and dynamic result than fixed text analysis. The free text analysis approach [3] is limited by the grouping problem. For example, if the considered character set {Q, W, E, R, T} is L1 group and {G, H, J, K, L} is R2 group then flight time for {HT} and {LQ} cannot be same.So assuming any pair in the (L1 ,L2) set in the same group will be erroneous. This grouping problem strategy can be solved if we consider a group that is formed with one's close neighbor who is just one step away. [1] had found very promising result in their experiment but the issues of scalability and sample size are not considered in their digraph approach.
Objective of proposed work is to analyse the keystroke database of different users and the discovery of a unique pattern for each user. This will further be extended to form the training data set which will be used for constructing the intelligent verifier to authenticate remote users from their continuous typing style. In order to achieve the objective, an approach is made to identify dense region of each group for a particular user.

3. PROPOSED ALGORITHM

The proposed solution consists of two phases viz. Data Collection Phase (Section 3.1) and Data Processing Phase (Section 3.2). Before moving to detailed description of proposed work, some terminologies are described as follows:

a. **Overlapping:** Overlapping is defined as the event of more than one key being pressed at the same time
b. **Range:** Range denotes time span between minimum and maximum flight time within which flight time for specific move resides. Some functions involving ranges are described below.
   . **Min_Range(r):** This is calculated as the minimum flight time of the range.
   . **Max_Range(r):** This is calculated as the maximum flight time of the range.

```
Procedure Distance (range i,range j)
If Max_Range(i)<Min_Range(j) then
        return (Max_Range[j]−Min_Range[i]);
Else
        return (Max_Range[i]−Min_Range[j]);
End If
```

```
Procedure Merge (range i,range j)
If Min_Range[j]> Max_Range[i] then
        Set min_range of range x= Min_Range[i]
        Set max_range of range x=Max_Range[j]
Else
        Set min_range of range x= Min_Range[j]
        Set max_range of range x=Max_Range[i]
End If
Return range x
```

c. **Occurrence:** This is defined as a total number of items present in range or group. It is denoted by Occr(x) where x is range or group.
d. **Diameter:** This is defined as the difference between maximum and minimum value of a particular range.
   . **Diameter(r) = Max_Range(r) − Min_Range(r)**

e. **Dense Area:** This is defined as the highest occurrence among several ranges for a particular group.It is denoted by following mathematical expression.
. **Dense Area = Max (Occr(X_i) | $X_i \epsilon$ Ranges of a particular group)**

```
Procedure Dense_Area(x)
For each range y of group x
        Find Occr(y)
End For
Find max occr(y | y ϵ group x)
```

f. **Training_Factor_Probability:** This is used to filter set of ranges to consider before merging the smaller sets. It is represented by the following mathematical notation:
. **TFP(x,y) = { x/y when y $\neq$ 0 | y ϵ occr(group i) and x ϵ occr(range j | j ϵ group i)}**

g. **Training_Factor_Ratio:** This helps to merge small diameter ranges to expand the range to give relaxation to a user typing style according to need of application. It is represented by the following mathematical notation:
. **TFR(x)= {Occr(x) : Diameter(x) | x is a range of a particular group}**
For different TFR a group can be split in different ranges. If narrow range is required then higher TFR is required to be set otherwise smaller TFR will reach the objective.

3.1. Data Collection Phase

An interface has been designed using JAVA Swing for collecting the typing data from users. The interface requires NetBeans 7.3, and the resulting database is stored in **.accdb** file. Once the data is collected, it is taken to the processing module for grouping.
Authors have selected research scholars (ages lie between 25-35) from our university for typing same long paragraph over three months at different instant of time. Users have used standard HP®wired USB desktop keyboard. Authors have collected 8 to 10 sample from each user. This dataset is remain private to our experiment.

3.2. Data Processing Phase

A static table HOP_TAB{num,x|xϵEnglish letters} 26 x 26 has been constructed which contains type of hops between each pair of keys. A snapshot of the table is shown in Table 1. Left most column (num) contains the number which actually designates alphabetsEnglish letters starting from A (=1) , B (=2) and so on. During enrollment phase jumps are ignored if they do not involve two characters. To categorize a jump from one key to another the table HOP_TAB is consulted .

3.2.1. Construction of HOP_TAB

The HOP_TAB is used to classify flight time. There are three rows in the keyboard which contains English letters. First row contains maximum columns (10) of letters. So authors assume representation of letters in keyboard as 3x10 matrix format. Thus, a jump operation can be described in terms of row movement and column movement. First step is to count row movements and then look for column movements. Movement can be either up or down or along the same row. The proposed solution allows up/down movements for maximum two steps i.e. from last row to first row and vice versa. Up and down movements are considered mirror image of each other. Second step is to consider the column movements. As first row

Table 1. HOP_TAB table showing shift requirements for jumping from one key to another.

-	ID	A	B	C	D	E
1. If only row operation is performed	1	SS	D1R4	D1R2	S0R2	U1R2
$\sum_{i=1}^{2}(U_i/D_i)$ = 2 groups	2	U1L4	SS	S0L2	U1L2	U2L2
2.If combination of row and column operations are performed.	3	U1L2	S0R2	SS	U1O	U2O
a. $\sum_{i=1}^{2}\left((U_i/D_i)\sum_{j=1}^{9}(L_j)\right)$ = 18 groups	4	S0L2	D1R2	D1O	SS	U1O
b. $\sum_{i=1}^{2}\left((U_i/D_i)\sum_{j=1}^{9}(R_j)\right)$ = 18 groups	5	D1L2	D2R2	D2O	D1O	SS
3. If only column operations is performed within same row	6	S0L3	D1R1	D1L1	S0L1	U1L1
a. $(S_0)\,(\sum_{i=1}^{9}(L_i) + \sum_{j=1}^{9}(R_j))$= 18 groups	7	S0L4	D1O	D1L2	S0L2	U1L2
b. $(S_0)(S_0)$ =1 group	8	S0L5	D1L1	D1L3	S0L3	U1L3
Total = 57 groups	9	D1L7	D2L3	D2L5	D1L5	S0L5
Fig. 1. Total combinations of shifting operation	10	S0L6	D1L2	D1L4	S0L4	U1L4

contains 10 English letters, maximum 9 left and 9 right operations are possible. Then row and column operations are combined. So, possible combinations are grouped in three parts and described in Figure 1. Here U/D represents up/down move and L/R represents left/right move and S represents same row move and SS represents same key press twice.

3.2.2. MODULE FOR DIGRAPH PROCESSING

```
For each consecutive English letters pair in raw database
 Consult HOP_TAB to identify what type of move
 Store flight time for the pair
       under that particular move in database
End For
```

3.2.3. THE ALGORITHM TO IDENTIFY THE DENSE REGION

```
For each group i Evaluate  total_length for the group i
 Divide each group vector into two  subgroups
 //Overlapping Vector and Non-overlapping Vector
 For each sub group j follow the steps
  divide each subgroup j  in ranges  of diameter 5ms
  sort ranges of each sub group j order by Occr(range)
  assign rank in ascending order
  evaluate allow_TFP= (TFP(range with rank 1,group i)/5)
  Assign ranges in variable  old_group
  add rank 1 range in variable new_group
  Do
   For each range  k=2 to n  in old_group perform following
    For each range m in new_group perform following
     If  TFP(range k,group i) > allow_TFP
      If distance(range m,range k)<= 10
       merge(range m,range k) in new_group
      Else
       add range k to new_group
      End If
```

```
      End If
    End For
  End For
  While (If any changes observed in new_group)
  sort ranges in new_group for sub group j order by Occr(range)
  assign rank in ascending order
  Range with rank 1 in new_group identifies dense area
 End For
End For
```

4. Result Analysis

It is observed that users make different mistakes at different time while typing. Authors do not treat mistakes separately rather they include it for experiment as separate key movement. Usually user key stroke flight time resides between -200ms to 500ms while a user is typing continuously. Each group is divided into two subgroups based on positive or negative flight time. As negative flight time indicates overlapping in typing, ranges are evaluated for finding out overlapping zone separately. A group of range is created from each subgroup of diameter 5ms and Occr (x) is calculated for each range x. Range is sorted depending on the occurrence and rank is assigned in ascending order. An effort is put on merging this range depending on two factors TFP(x,y)and TFR(x). First range is judged on the training probability factor. Training probability factor is allowed to 5 times down the TFP(x,y) of the range x whose rank is 1. It helps to remove some range which are lying below the allowed probability. Then TFR(x) is applied to merge small diameters iteratively depending on the requirement.
The Occr(group) vs. sequence number is plotted and range evaluation of the proposed work is presented. Two groups with respect to two users are compared in the above Figure 1, where graphical plot has shown a clear visual distinction of two users' key stroke pattern for same row movement with left shift 1 and 2. While processing raw data, if occurrence of a particular group (say, same row left 1 move) is found, then it is stored in a table as (sequence no, flight time).When first item is found its sequence number is 1 and when second item with same move is found its sequence number is 2 and so on. Here cross section axis style gives a clear difference between overlapping and non overlapping area. Flight time is plotted against x axis whereas sequence number has plotted against y axis. Right side of Y axis designates non overlapping area whereas left side indicates overlapping characteristic in typing rhythm of a user. Dense area is clearly observed in the Figure 2 and Figure 4 whose numeric data are mentioned in Table 2 and Table 4 respectively. Similarly, Figure 3 and Figure 5 depicts dense area prominently for same row left 2 movement whose numeric data is tabulated in Table 3 and Table 5. Some other ranges with less TFP are also shown in the tabular form which is also different for two different users. Here in Table 6 some groups from 19 different possible moves in same row movement is presented. Authors have shown only dense area for a particular move. Finally authors have plotted Probability of Occr(x) vs Range x where diameter of range is 10ms i.e. overlapping zone (lying between -200 to 0) is divided in 20ranges whereas non-overlapping zone (lying between 0 to 500) is divided in 50 ranges. Authors have used Max(Range) in the plot to depict the range. For example to depict the Range(10,20) authors have shown 20 along x axis. Authors have plotted same row left 1 movement in Figure 6 and same row left 2 movement in Fig 7 for 5 users. Different color curve represents different user and every user has same color in two plots. Plots clearly show difference in users' typing pattern and can be treated as characteristics of a user.

Table 2. Result of User1 for SL1 movement.

ID	NAME	MIN	MAX	TFP
1	S L 1	-95	-45	0.5422
2	S L 1	75	145	0.248
TFR = 0.7				
ID	NAME	MIN	MAX	TFP
1	S L 1	-95	-45	0.5422
2	S L 1	75	95	0.1194
3	S L 1	105	115	0.0547
4	S L 1	140	145	4.48E-02
5	S L 1	125	130	2.99E-02
6	S L 1	30	35	2.49E-02
7	S L 1	185	190	2.49E-02
TFR = 0.9				

Table 3. Result of User1 for SL2 movement

ID	NAME	MIN	MAX	TFP
1	S L 2	125	175	0.4958
2	S L 2	30	50	8.26E-02
3	S L 2	-49	-29	5.79E-02
4	S L 2	75	95	7.85E-02
TFR 0.7				
ID	NAME	MIN	MAX	TFP
1	S L 2	125	175	0.4958
2	S L 2	30	50	8.26E-02
3	S L 2	75	95	7.85E-02
4	S L 2	-49	-44	3.31E-02
5	S L 2	-19	-15	3.31E-02
6	S L 2	-34	-29	2.48E-02
TFR 0.9				

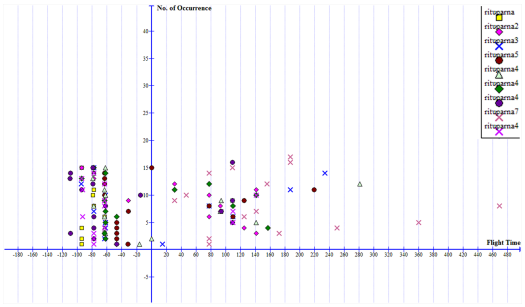

Fig. 3. Result of User1 for same row left−1 movement.

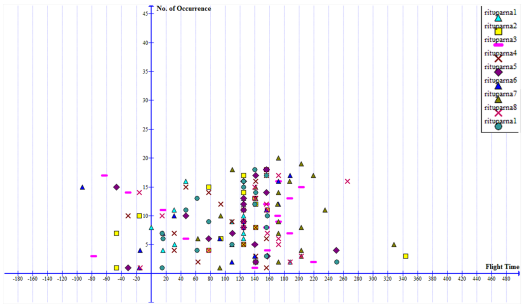

Fig. 4. Result of User2 for same row left−2 movement.

Table 4. Result of User2 for SL1 movement.

ID	NAME	MIN	MAX	TFP
1	SL1	15	65	0.3478
2	SL1	-34	-15	0.2319
3	SL1	75	95	0.1594
4	SL1	-79	-74	5.80E-02
TFR = 0.7				
ID	NAME	MIN	MAX	TFP
1	SL1	15	65	0.3478
2	SL1	-34	-15	0.2319
3	SL1	75	95	0.1594
4	SL1	-79	-74	5.80E-02
TFR = 0.9				

Table 5. Result of User2 for SL2 movement.

ID	NAME	MIN	MAX	TFP
1	SL2	-95	-60	0.494318
2	SL2	15	80	0.301136
3	SL2	-20	-15	8.52E-02
TFR 0.7				
ID	NAME	MIN	MAX	TFP
1	SL2	-95	-60	0.494318
2	SL2	15	80	0.301136
3	SL2	-20	-15	8.52E-02
TFR 0.9				

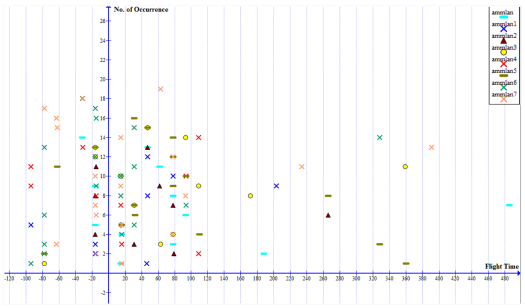

Fig. 4. Result of User2 for same row left−1 movement.

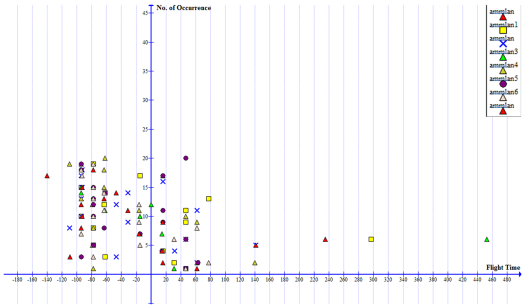

Fig. 5. Result of User2 for same row left−2 movement.

5. Conclusion

The authentication of remote user occupies a vital role in securing real life applications such as remote healthcare, banking, etc. The collection of user typing profile is a popular biometrics form of authentication due to its ease of implementation. However, in order to generate accurate keystrokes profile of any user, proper grouping is very important. This paper

Table 6. Different Type of Moves for User1 and User2.

NAME OF MOVE	User1	User2
SL4	(121,126,0.125), (171,176,0.125)	(46,51,0.212)
SL5	(15,110,0.386)	(15,80,0.558)
SL6	(156,161,0.174)	(31,36,0.2)
SL7	(45,50,0.132)	(91,96,0.235)
SR1	(-126,-46,0.459)	(15,65,0.354)
SR2	(76,191,0.646)	(30,65,0.279)
SR3	(-112,-32,0.502)	(15,80,0.729)
SR4	(-80,-15,0.525)	(15,80,0.511)
SR5	(-66,-61,0.128)	(15,35,0.39)
SR6	(-49,-15,0.329)	(-33,-15,0.3125)
SR7	(90,95,0.125)	(15,20,0.135)

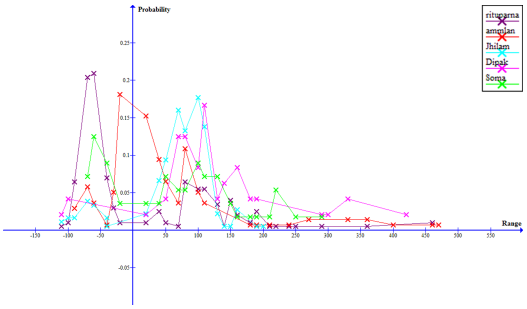

Fig. 6. Plot of 5 users for same row left 1 movement.

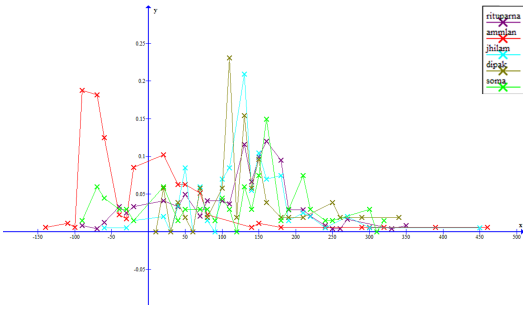

Fig. 7. Plot of 5 users for same row left 2 movement.

considers grouping of user keystrokes to find a unique pattern. Groups for same row movement have been examined and resulted in the identification of dense area for each user. It is observed that maximum number of groups are filled up in enrollment phase as group size is low (57) compared to the approach [1]. Their approach had to fill up the entire 26 X 26 matrix, if it is assumed that they worked with English letters only. Authors are now working on creation of a larger user database for comparing this logic with other existing techniques in this domain.

BIBLIOGRAPHY

[1] AHMED A. A., TRAORE I., Biometric Recognition Based on Free-Text Keystroke Dynamics, IEEE TRANSACTIONS ON CYBERNETICS, 2014, pp. 458-472.

[2] RYBNIK M., PANASIUK P., SAEED K., ROGOWSKI M., Advances in the Keystroke Dynamics: The Practical Impact of Database Quality, IFIP International Federation for Information Processing, 2012, pp. 203-214.

[3] SINGH S., ARYA DR. K. V. , Key Classification: A New Approach in Free Text Keystroke Authentication System,Circuits,Communication & System(PACCS), 2001, pp. 1-5.

[4] MONROSE F., RUBIN A. D. , Keystroke dynamics as a biometric for authentication, Future Generation Computer Systems, 2000, pp. 351-359.

[5] RYBNIK M., TABEDZKI M., SAEED K. , Keystroke Dynamics Based System for User Identification, 7th computer Information System & Management Applications, 2008, pp. 225-230.

[6] An augmented computer user login authentication using classifying regions of keystroke density neural network. Patent No: PST-15418/36 40410sh

[7] RYBNIK M., PANASIUK P., SAEED K., User Authentication with Keystroke Dynamics using Fixed Text",International Conference on Biometric and Kansei Engineering, 2009, pp. 70-75.

[8] ROBINSON J. A., LIANG V. M., CHAMBERS J. A. M., MACKENZIE C. L., Computer User Verification Using Login String Keystroke Dynamics, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART A: SYSTEMS AND HUMANS, 1998, Vol. 28, No. 2, pp. 236-241.

[9] SIM T., JANAKIRAMAN R., Are Digraphs Good for Free-Text Keystroke Dynamics?, Computer Vision and Pattern Recognition CVPR '07, 2007 , pp. 1-6.

[10] MR N. PAVADAY, SOYJAUDAH DR. K. M. S., Investigating performance of Neural Networks in authentication using keystroke dynamics, AFRICON, 2007, pp. 1-8.