Enhanced Continuous User Authentication Based on Specific Keystroke Features

Akash Sanghi, Y.D.S. Arya

Invertis University, Bareilly

ABSTRACT- User authentication is one of the basic necessities for any security system. Identifying an individual based on a username, password or any other means ensures that the person is same who he or she claims to be when accessing a system, application or network. We need methods to prevent unauthorized access to critical business data, but traditional authentication systems are not enough to provide strong security throughout a user work session. That's where continuous authentication is required. Continuous authentication is a dynamic authentication that examines attributes which changes more than the acceptable range and continually looks to validate the current user.

An approach for continuous authentication is implemented based on the specific keystroke features of an individual. Our proposed approach, Continuous User Authentication Using Specific Keyset (CUAUSK) Algorithm outperforms the approach which is based on the keystroke behavior of users without considering the specific keystroke patterns.

KEYWORDS: *Authentication, Keystroke Dynamics, Flight time, Specific timing feature, Significant timing feature, General user behavior, Confidence value, Score value, Threshold.*

1. INTRODUCTION

Biometric systems automatically recognize individuals based on their physiological and/or behavioral characteristics like fingerprint, face, hand-geometry, iris, retina, palm print, voice, gait, signature, and keystroke dynamics. Keystroke patterns (typing patterns) are a recognized behavioral biometric for establishing the security credentials of users in the context of static user authentication. The fundamental idea is that the rhythm of typing a predefined text by a legitimate user can be learned, and consequently used for authentication purposes. However, there is one disadvantage that all static authentication methods share. They authenticate the user at the moment that the authentication mechanism is executed: any change of user after that will be unnoticeable to the system. A completely different type of authentication is continuous authentication [13], which is used after an (authenticated) user has entered a system. The system will then continuously monitor if a change of user occurs. Not every authentication method can be used for continuous authentication. In particular, we are restricted to biometric methods, where again we will be restricted. Biometric features like fingerprints or iris scans are not suitable for continuous authentication on a computer. Keystroke dynamics is generally used for continuous authentication as keystroke patterns of an individual cannot be easily replicated or stolen by an impostor, even if such patterns are known. The continuous authentication system based on keystroke dynamics will lock out a user if the trust in genuineness of the current user is too low. Ideally such a system would never lock out a genuine user and detect an impostor user within as few keystroke actions as possible.

This paper proposes an algorithm based on keystroke dynamics. Concept of specific timing features is introduced, which suggests that every user's behavior is specific for some particular keys. For those keysets, user behaves much differently than other general users. System exploits this particular characteristic of users to calculate the score value which will be the basis for the confidence level to suggest whether the current user is the same genuine user or has changed during a single session.

The paper is organized as follows: Section 2 discusses the related work. Section3 gives an insight into the background knowledge required to understand the terminologies used in the paper. Section4 explains the proposed algorithm. Section5 outlines the experimental setup and obtained results. Finally Section 6 concludes the topic.

2. RELATED WORK

Sulong et al. [3] have proposed a system combining maximum pressured applied on the keyboard and latency between keystrokes as input to a radial basis function network. They achieved 100 % classification rate with 22.4 s average training time. Based on FRR and FAR, the authors claimed that the proposed system is effective for biometric-based security systems.

B. Draffin et al. [4] performed experiments utilizing soft keyboard data collected from 13 participants over 3 weeks. The study used key-press duration, finger area, drift, pressure, and keyboard orientation as features, and achieved a 14 % FAR and 2.2 % FRR.

H. Saevanee et al. [6] studied timing features combined with finger pressure and used notebooks with touchpads as a touchscreen. Data was collected from 10 users, who entered their 10-digit cell phone number. The experiment yielded 99 % accuracy using the finger pressure features. A limitation of this approach is lack of impostor data due to each participant having a different phone number. In this case, only FRR was measured.

R. A. Maxion et al. [17] conducted an experiment where 28 users typed the same 10 digit number using only the right-hand index finger. The authors used a random forest classifier and have achieved a 10 % EER.

Robert S. Zackt et al. [18] developed a long-text input keystroke biometric system that consists of three components: raw keystroke data collection over the Internet, a feature extractor, and a pattern classifier. The system was tested with 120 participants and achieved approximately 1 % EER. The system showed higher performance with a closed system of known users than an open system, as well as performance variations with the number of enrolled users.

S. Sen et al. [19] performed a study which used pressure as a feature, with 4-digit input from 10 participants. The study presented verification results based on a special impostor mode in addition to the typical performance measures.

T. Samura and Nishimura [21] conducted a study that examined keystroke dynamics for long free-texts. The experiment participants were divided into three groups based on their typing speed, specifically the number of letters typed in a 5 minute period. This study indicated that the best recognition accuracy was obtained from the group which typed fastest.

Y. Deng et al. [22] have introduced two new algorithms: Gaussian mixture model with universal background model (GMM-UBM) and deep belief nets (DBN). These two approaches leverage data from background users and enhance the model's discriminative capability without using imposter's data at training time. The authors claimed that these two new algorithms make no assumption about underlying probability distribution and are fast for training and testing.

3. BACKGROUND KNOWLEDGE

In this section, some of the terminologies [2] used for understanding the proposed algorithm are described. These terms are used later in this paper at various points.

1. **Keyset:** Keyset is defined as combination of any two keys pressed by a user e.g. th. System will capture the time interval from release of key "t" and pressing of key "h".

2. **Flight Time:** The time interval between a key release and the next key press.

3. **General User Behavior (μ_{avg}) :** It represents the mean flight time values of n users for each keyset. It is also referred as average set or mean behavior.

4. **Specific Keyset:** This is a set of "x" keysets where user's flight time behavior is maximum distant from average set. This can be found out by sorting the user's deviation table and selecting the topmost "x" values.

5. **Significant Keyset:** This is a set of next "x" keysets where user's flight time distance from average set is less then specific keyset and greater than the normal keyset.

6. **Normal Keyset:** All other keysets which are neither in specific category nor in significant category are termed as normal keysets for an individual user.

7. **Deviation (d):** Deviation of any keyset from the stored template is the time difference between the current value and the stored flight time value for that particular keyset.

8. **Acceptable Range (R):** This is defined as the range of values which are allowed to deviate from the stored template in order to still qualify as accepted input keyset for a particular keyset entered.

9. **Penalty value (c):** This is the value with which system decrements or increments the score value S, depending upon whether the deviation "d" is acceptable or not, in case of normal keyset input by user.

10. **Confidence Score Value (S):** This is the value which is calculated in order to determine whether system should allow the present user to keep working on the system or ask to go through the logon process again, in order to prove the identity.

11. **Critical Threshold value ($T_{critical}$):** This value reflects the threshold level, which if reached will signify that the user is probably an imposter and should be logged out of the system immediately.

4. CONTINUOUS USER AUTHENTICATION USING SPECIFIC KEYSET ALGORITHM (CUAUSK)

This algorithm exploits the specific keystroke behavior of individual user in the authentication process. Algorithm is divided into two phases:

1. Data Acquisition and Optimization
2. Continuous Authentication

Phase 1: Data Acquisition and Optimization

This phase comprises of five steps which are as follows:

Step 1 (Data acquisition: Flight time values)

- Input keysets for user
- Take average for duplicate keyset values (if any)
- Store these average flight time values in user table (say column A)

Repeat step1 for each of the 'n' users 'm' times for the same keysets.

Step 2 (Mean value calculation for each user)

- Generate mean flight time values for every input keyset of each of the n users.
- Relative to every user, generate mean value tables as a_1, a_2 ……. a_n.

Step 3 (Mean flight time calculation μ_{avg})

- Take mean flight time values of all n user's mean value tables for the corresponding keysets.
- Store mean flight time calculation table as μ_{avg} table.

$$\mu_{avg} = \frac{1}{n}\left(\sum a_n\right)$$

Step 4 (Calculate user's deviation from mean flight time values)

- Take deviation of mean flight time values of each user from μ_{avg} table
- Store deviation of each user in dev_a$_n$ table.

$$dev\_a_n = (|\mu_{avg} - a_n|)$$

Step 5 (Optimize user's template)

- Sort the deviation tables of all user's, individually from high to low to get specific (represented by topmost 'x' values) and significant keysets (represented by next 'x' values) for each user.

Phase 2: Continuous Authentication

This phase comprises of two steps which are as follows:

Step 1 (Confidence Score Calculation)

Initially set S=0;

if input key set is not available in the stored template for the particular user then

$$S = S + \alpha;$$

else if the input keyset is in specific category for the particular user then

$$S = \begin{cases} \max(0, S - 3c) & \text{if } d \leq R; \\ S + 3c & \text{if } d > R; \end{cases}$$

else if the input keyset is in significant category for the particular user then

$$S = \begin{cases} \max(0, S - 2c) & \text{if } d \leq R; \\ S + 2c & \text{if } d > R; \end{cases}$$

else if the input keyset is in normal category for the particular user then

$$S = \begin{cases} \max(0, S - c) & \text{if } d \leq R; \\ S + c & \text{if } d > R; \end{cases}$$

For each keyset entered we get in return updated score value S.

Step 2 (Comparison and Action)

Compare S and $T_{critical}$ at each updation of S value.

if S< $T_{critical}$ then continue use of system

else

logout user

Parameters Used

On performing experiments for n=25 and m=10, using the above algorithm, established and optimized parameter values are as follows:

α = 0.01, x= $\lfloor$5% of total input keysets$\rfloor$, R= 97ms, c=0.08 and $T_{critical}$= 15.5

5. EXPERIMENTAL SETUP AND RESULTS

An experiment was conducted on 25 engineering students to analyze the behavior of algorithm. Data acquisition was spanned over a time of 15 days, so that we can get general keystroke behavior of each user. The dataset consist of 418 keysets. System was implemented using NetBeans IDE for java and MS SQL Server. Figure 1(a) shows the data acquisition module of the system and figure 1(b) shows the flight time values for the entered keysets.

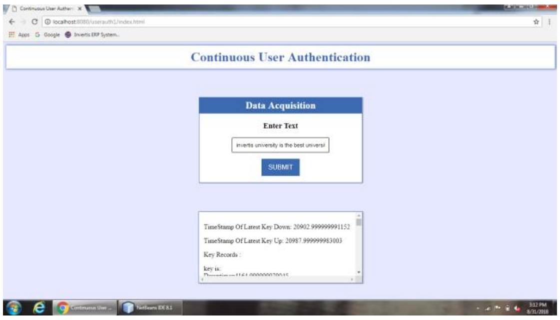

Figure 1(a): Data Acquisition Module

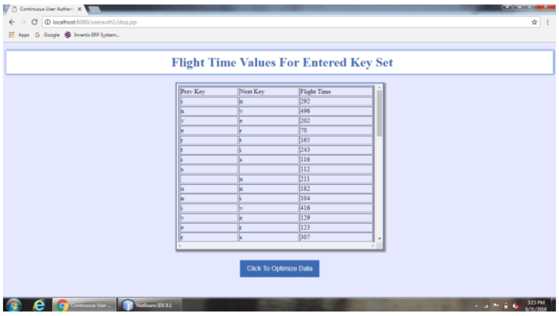

Figure 1(b): Flight Time Values

These flight time values are optimized and stored into database. Figure 2(a) shows the optimization module and figure 2(b) reflects the stored values in the database.

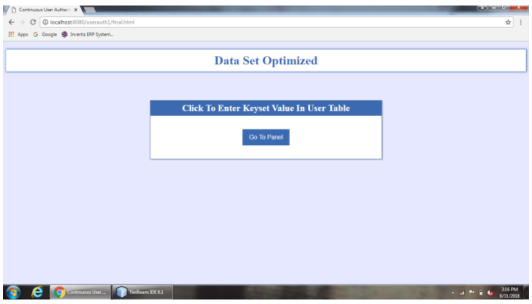

Figure 2(a): Data Set Optimized

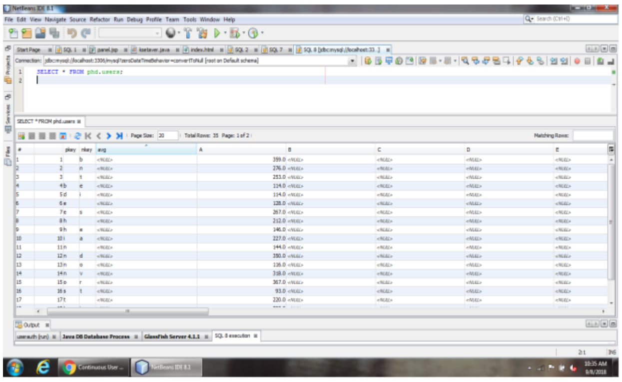

Figure 2(b): User Value in database

Through panel module, navigation can be performed to the other functionalities of the system, which is shown in figure 3(a). Figure 3(b) reflects the entry section for continuous authentication module.

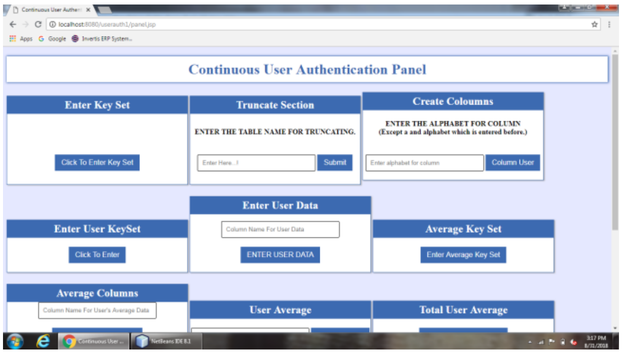

Figure 3(a): Panel Module

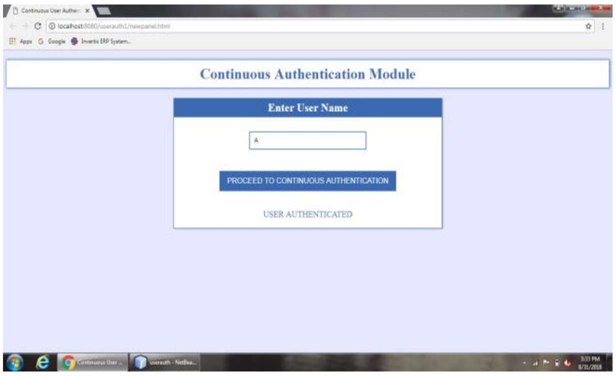

Figure 3(b): Continuous Authentication (Entry Section)

User is allowed to continually use the system as far as the trust level is acceptable i.e. the score value (S) is below the defined threshold. Figure 4(a) shows the continuous authentication process along with the current keysets entered in real time. Figure 4(b) reflects the action taken by the proposed system, as soon as the trust level falls below the threshold level reflecting the suspicion of an intruder. As the score value reaches the defined threshold, user is logged out of system and asked to login again.

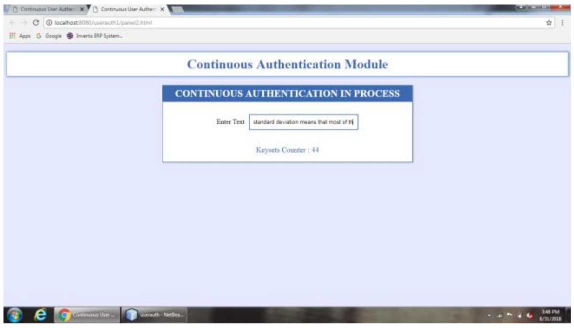

Figure 4(a): Ongoing Continuous

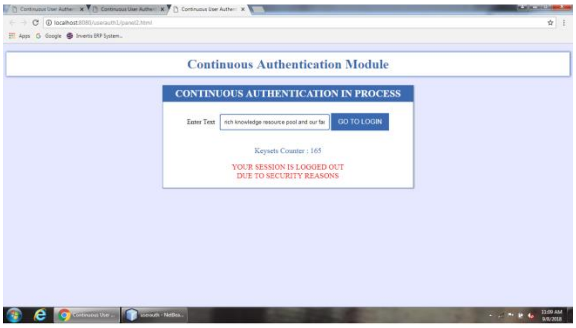

Figure 4(b): Action on Intruder Authentication

Results obtained on performing the experiments using the proposed algorithm (Case 1) and without considering the specific keyset criteria (Case 2) as suggested in the algorithm are as follows:

CASE 1: System considers Specific keyset features (according to CUAUSK algorithm)

Keystroke behavior of two different intruders (Intruder 1 and Intruder 2) was analyzed in the login of every registered user. Results indicate the number of keysets entered by the intruder in the system of every user before being logged out of the system. Mean value of lockout keystrokes are found to be 173.52 and 166.24 for Intruder 1 and Intruder 2 respectively. Figure 5 presents the results obtained and table 1 shows the lockout keystroke values for Intruder 1.

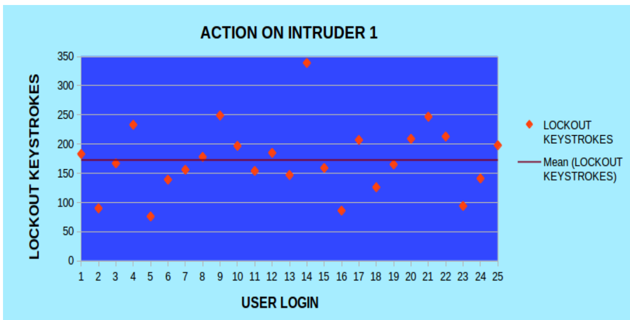

Figure 5: Action on Intruder 1 (using CUAUSK)

Table 1: Lockout keystroke values for Intruder 1

USER	LOCKOUT KEYSTROKES	USER	LOCKOUT KEYSTROKES
1	183	14	339
2	90	15	159
3	167	16	86
4	233	17	207
5	76	18	126
6	139	19	165
7	156	20	209
8	178	21	247
9	249	22	213
10	197	23	94
11	154	24	141
12	185	25	198
13	147		

Figure 6 presents the results obtained and table 2 shows the lockout keystroke values for Intruder 2.

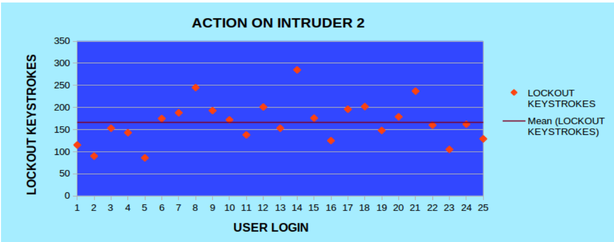

Figure 6: Action on Intruder 2 (using CUAUSK)

Table 2: Lockout keystroke values for Intruder 2

USER	LOCKOUT KEYSTROKES	USER	LOCKOUT KEYSTROKES
1	115	14	285
2	90	15	176
3	153	16	125
4	143	17	196
5	86	18	202
6	175	19	148
7	188	20	179
8	245	21	237
9	193	22	160
10	172	23	105
11	138	24	162
12	201	25	129
13	153		

It is always possible that the user working on the system is actually an authentic user. When two authentic users worked on the system the results obtained are shown in table 3.

Table 3: Lockout keystroke values for authentic users

USER	LOCKOUT KEYSTROKES
Authentic User 1	8756
Authentic User 2	9831

These results indicate fairly high values of lockout keystrokes which suggest that an authentic user can continue to work on the system for sufficient duration during a logon session.

CASE 2: System do not consider Specific keyset features (act according to normal keystrokes only for all keysets)

Same intruders (Intruder 1 and Intruder 2) worked in the login of every registered user. Results indicate the number of keysets entered by the intruder in the system of every user before being logged out of the system. Mean value of lockout keystrokes are found to be 322.96 and 310.04 for Intruder 1 and Intruder 2 respectively, which are much higher as compared to the system based on the proposed algorithm. Figure 7 presents the results obtained and Table 4 shows the lockout keystroke values for Intruder 1.

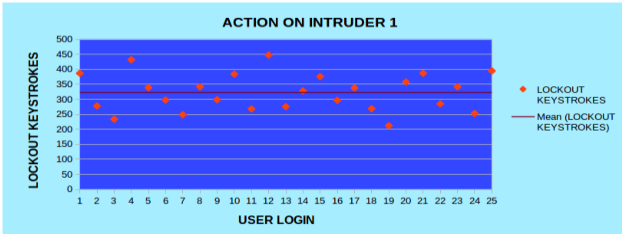

Figure 7: Action on Intruder 1 (without using CUAUSK)

Table 4: Lockout keystroke values for Intruder 1

USER	LOCKOUT KEYSTROKES	USER	LOCKOUT KEYSTROKES
1	387	14	328
2	278	15	376
3	234	16	297
4	432	17	338
5	339	18	269
6	298	19	212
7	249	20	358
8	342	21	387
9	299	22	285
10	384	23	342
11	268	24	253
12	448	25	395
13	276		

Figure 8 presents the results obtained and Table 5 shows the lockout keystroke values for Intruder 2.

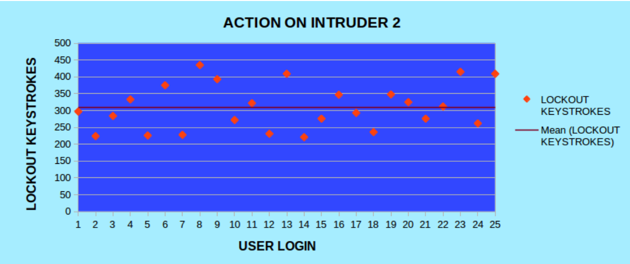

Figure 8: Action on Intruder 2 (without using CUAUSK)

Table 5: Lockout keystroke values for Intruder 2

USER	LOCKOUT KEYSTROKES	USER	LOCKOUT KEYSTROKES
1	297	14	221
2	224	15	276
3	284	16	347
4	333	17	293
5	226	18	236
6	375	19	348
7	228	20	325
8	435	21	276
9	393	22	312
10	272	23	415
11	322	24	262
12	231	25	411
13	409		

When same authentic users (as in case 1) worked on the system the results obtained are shown in table 6.

Table 6: Lockout keystroke values for authentic users

USER	LOCKOUT KEYSTROKES
Authentic User 1	7625
Authentic User 2	8339

These results are less effective as compared to the system based on the proposed algorithm.

5. CONCLUSION

The results obtained by considering the specific keystrokes of any user for authentication are fairly impressive and outperforms the system which does not assign weight to specific keystroke features of any user. Thus a system which considers the specific patterns of individual user may prove to be more efficient and can definitely enhance the results. It performs better in both the scenarios, i.e. whether it is an intruder or an authentic user.

REFERENCES

[1] A.Darabseh & A.Siami Namin, "Keystroke active authentications based on most frequently used words", in Proceedings of the ACM international workshop on international workshop on security and privacy analytics (pp. 49–54), 2015.

[2] A.Sanghi & Y.D.S. Arya, "A Novel Approach for Continuous User Authentication using Keystroke Dynamics**"** International Journal of Engineering Technology Science and Research, Volume 4, Issue 10, October 2017

[3] A.Sulong, W.Wahyudi & M.Siddiqi, "Intelligent keystroke pressure-based typing biometrics authentication system using radial basis function network", in Proceedings of the 5th international colloquium on signal processing and its applications (CSPA '09) (pp. 151–155), 2009.

[4] B.Draffin, J.Zhu & J.Zhang, "KeySens: passive user authentication through micro-behavior modeling of soft keyboard interaction",in Springer International Publishing (Vol. 130), 2011.

[5] H.Ali, W.Wahyudi & M. Salami, "Keystroke pressure based typing biometrics authentication system by combining ann and anfis-based classifiers", in Proceedings of the 5th international colloquium on signal processing and its applications (CSPA '09) (Vol. 1, pp. 198–203). Kuala Lumpur, 2009.

[6] H.Saevanee & P.Bhattarakosol, "Authenticating user using keystroke dynamics and finger pressure", in 6th IEEE consumer communications and networking conference (CCNC'09) (pp. 1–2). Las Vegas, 2011.

[7] J.Monaco, M.Ali, & C.Tappert, "Spoofing key-press latencies with a generative keystroke dynamics model", in 7th international conference on biometrics: theory, applications and systems (BTAS), 2015.

[8] J.Roth, X.Liu, A.Ross & D.Metaxas, "Biometric authentication via keystroke sound", in international conference on biometrics (ICB) (pp. 1–8), 2013.

[9] K.Balagani, V.Phoha, A.Ray & S.Phoha, "On the discriminability of keystroke feature vectors used in fixed text keystroke authentication", in Pattern Recognition Letters, 32(7),(pp. 1070–1080), 2011.

[10] M.Ali , J.Monaco , C.Tappert , & M.Qiu . "Authentication and identification methods used in keystroke biometric systems", in IEEE international symposium on big data security on cloud (BigDataSecurity) (pp. 1424–1429), 2015.

[11] M.Antal , L.Szabo & I.Laszlo, " Keystroke dynamics on android platform", in INTER-ENG , 8th international conference interdisciplinary in engineering (pp. 114–119). Tirgu Mures: Elsevier, 2008.

[12] M.El-Abed, M.Dafer & R.El Khayat, “Keystroke: a mobile-based benchmark for keystroke dynamics systems”, in International Carnahan conference on security technology (ICCST) (pp. 1–4), 2014.

[13] M.Rybnik, M.Tabedzki, M.Adamski & K.Saeed, “An exploration of keystroke dynamics authentication using nonfixed text of various length”, in international conference on biometrics and Kansei engineering (ICBAKE) (pp. 245–250), 2013.

[14] N.Bakelman, J.Monaco, S.Cha & C.Tapper, “Keystroke biometric studies on password and numeric keypad input”, in European intelligence and security informatics conference (EISIC) (pp. 204–207), 2013.

[15] P.Bours & E.Masoudian, “Applying keystroke dynamics on one-time pin codes”, in International workshop on biometrics and forensics (IWBF) (pp. 1–6), 2014.

[16] R.Giot, M.El-Aed & C.Rosenberger, “Web-based benchmark for keystroke dynamics biometric systems: a statistical analysis”, in eighth international conference on intelligent information hiding and multimedia signal processing (IIH-MSP) (pp. 11–15), 2014.

[17] R.Maxion & K.Killourhy, “Keystroke biometrics with number-pad input”, in Proceedings of the IEEE/IFIP international conference on dependable systems and networks (DSN ’10) (pp. 201–210), 2010.

[18] R.Zack, C.Tappert & S.Cha, “Performance of a long-text-input keystroke biometric authentication system using an improved k-nearest-neighbor classification method”, in Fourth IEEE international conference on biometrics: theory applications and systems (BTAS) (pp. 1–6). Washington DC, 2009.

[19]. S.Sen & K.Muralidharan, “Putting pressure on mobile authentication”, in Seventh international conference on mobile computing and ubiquitous networking (ICMU) (pp. 56– 61). IEEE, Singapore, 2015.

[20] S.Singh & M.Sinha, “Pattern construction by extracting user specific features in keystroke authentication system”, in 4th international conference on computer and communication technology (ICCCT) (pp. 181–184), 2013.

[21] T. Samura and H. Nishimura, “Keystroke Timing Analysis for Individual Identification in Japanese Free Text Typing”, in ICROS-SICE International Joint Conference, 2009.

[22] Y.Deng & Y. Zhong, “Keystroke dynamics user authentication based on gaussian mixture model and deep belief nets”,in ISRN Signal Processing, 2013.