Enhanced User Authentication through Trajectory Clustering

Hazarath Munaga¹, J. V. R. Murthy¹, and N. B. Venkateswarlu²
¹Dept. of CSE, JNTU Kakinada, India
²Dept. of CSE, AITAM, Tekkali, India
Email: {hazarath.munaga, mjonnalagedda}@gmail.com
Email: venkat_ritch@yahoo.com

Abstract— Password authentication is the most commonly used technique to authenticate the user validity. However, due to its simplicity, it is vulnerable to pseudo attacks. It can be enhanced using various biometric techniques such as thumb impression, finger movement, eye movement etc. In this paper, we concentrate on the most economic technique, based on the user habitual rhythm pattern i.e. not what they type but how they type is the measure for authenticating the user. We consider the latency between key events as the trajectory, and trajectory clustering is used to obtain the hidden patterns of the user. Obtained pattern can be considered as a cluster of measurements that can be used to differentiate from other users. We evaluated the proposed technique on the data obtained from the 100 users.

Index Terms—key stroke analysis, trajectory clustering, key stroke latency, key stroke biometrics

I. INTRODUCTION

As a computer engineer, the main goal is to protect the information or resources from unauthorized users. In literature there are several methods to authenticate validity of the user varying from the usage of biometrics to smart cards. Among them, password authentication is the most acceptable and widely used mechanism because of its economical, operational and implementation advantages. The method relies on the fact that only the authorized user knows the correct password. There is no security in the use of passwords if an impostor knows the password. Hence, to improve the security of user authentication one option is to replace the passwords with a biometrics identification of the user. Currently, there are three major forms of biometrics: physiological, behavioral, and token based. Physiological based biometrics rely on biological attributes such as fingerprints, iris, and retina patterns. Behavioral biometrics utilize behavioral attributes such as voice, signature, and keystroke dynamics. Token-based systems require the possession of a security device such as an ID card. Behavioral biometrics works on the way we interact with the authentication system.

The most popular systems are based on voice, signature, and keystroke dynamics, out of these, keystroke dynamics is purely software-based, it is less expensive and more user transparent. As Gaines et al. [1] observation, a user’s keystroke pattern is highly repeatable and distinct from that of other users (typing biometric), which can be used to discriminate the owner from impostors. Hence, typing biometrics based authentication uses an individual’s unique typing pattern to validate the authentic user among impostors. The action of typing the password can be analyzed with respect to its physiological characteristics i.e. the latency time between keystrokes, keystroke pressure, key displacement, and key displacement duration.

Ref. [1] introduced the use of keystroke timings as a means of authentication using seven professional typists. Since then there have been a number of research studies [2-9] on authentication of users based on keystroke timings using various techniques ranging from deterministic algorithms to machine learning, and clustering algorithms as learning algorithms for classification.

Ref. [2] develops a profile using statistical analysis method involving means and standard deviations of latencies between consecutive keystrokes. Under these types of statistical models, if a user were to type each key much faster than usual, then he would most likely be rejected because the timing measurement of each of his pairs of consecutive keystrokes would fall beyond the stored mean of his trained profile.

In 1997 Monrose and Rubin use the Euclidean Distance and probabilistic calculations based on the assumption that the latency times for one-digraph exhibits a Normal Distribution [3]. Afterwards, in 2000, they also present an algorithm for identification, based on the similarity models of Bayes, and in 2001 they present an algorithm that uses polynomials and vector spaces to generate complex passwords from a simple one, using the keystroke pattern [4].

In 2000 [5] demonstrated using neural network (NN) novelty detection model, which was built by training the owner’s patterns only, and the model was used to detect impostors using some sort of similarity measure, reporting a 1.0% false rejection rate (hereinafter, FRR) and 0% false acceptance rate (hereinafter, FAR). Sung et al [6] has also applied NN to keystroke dynamics, generating error rates on the order of 2-4%. However, such solution suffers from typical NN limitations, e.g.,
conditional independence (i.e., being in a state depends only on the previous state).

Revet et al. have used the rough sets induction algorithm to extract rules that form models for predicting the validity of a login ID/password attempt [7]. The results indicate that the error rate can be as low as 2% in many cases. In addition, the use of a multiple sequence alignment algorithm has been successfully deployed to authenticate a group of users with virtually 100% success [8].

Ref. [9] demonstrated the usage of k-means clustering for validating the user using key stroke dynamics. Whereas, the k-means (i) is not suitable for generating non-globular clusters and, (ii) it is difficult to compare the quality of the computed clusters (e.g. the different initial partitions and the k value affect the outcome). Finally, the k-means approach requires lots of computational time for convergence.

The use of key stroke dynamics based user authentication is no longer a novel concept, but, novelty in this study is the adopted dissimilarity measure and the technique used to identify the user from impostors.

Ref. [11] demonstrated the usage of trajectory clustering for visualizing, analyzing and obtaining hidden patterns from user navigations obtained from virtual environments. Ref. [12] demonstrated the usage of trajectory clustering for selecting cluster heads which implicitly used to extenuate the life time of wireless sensor networks. In this study, we employ a trajectory based clustering algorithm for authenticate the legitimate user based on the two explicit (key pressed, key released) and one implicit (key typed, which is used to obtain the applied pressure on the key) events.

II. TRAJECTORY CLUSTERING ALGORITHM

The success of any clustering algorithm depends on the adopted dissimilarity measure. This section explains about the adopted dissimilarity measure.

Ref. [13], proposed the usage of Euclidean distance between time series of equal length as the measure of their similarity. The idea has been generalized in [14] for subsequence matching. In a similar way [15] used Discrete Wavelet Transform and [16] used Principal Component Analysis for measuring time series similarity.

Another approach which is brought from image processing is time warping technique and it is used in [17] to match signals in speech recognition. Berndt and Clifford [18] suggested this technique to measure the similarity of time-series data in data mining. Recent works have also used this similarity measure [19][20].

Ref. [9] suggested the usage of Canberra distance for finding the distance between user samples. Here we used Hausdorff measure [21] for calculating dissimilarity between trajectories, and observe that the Hausdorff measure is more sensitive for small changes than the Canberra distance, which we can observe clearly from the Fig. 1 (obviously, trajectories A and B are overlapped due to its similar nature). Note that, we consider the key stroke latency id also for calculating the Hausdorff dissimilarity.

The following are the some definitions used in our algorithm.

Definition 1 A trajectory \((t)\) is represented as \(trj(t) = (u_0, u_1, u_2, \ldots, u_n)\), where \(t\) is a unique trajectory id (user id), and \(<u_0, u_1, u_2, \ldots, u_n>_t >\) is a sequence of key events reflecting the key stroke latencies.

Definition 2 We define the key stroke dissimilarity function between two trajectories \(t_1\) and \(t_2\) as the maximum of one way distances between two trajectories.

As mentioned in [12], the one way distance from a trajectory \(t_1\) to another trajectory \(t_2\) is defined as the integral of the Hausdorff distance between points of \(t_1\) to trajectory \(t_2\) divided by the number of points \(|t_1|\) in \(t_1\).

\[
\text{dist}_{ow}(t_1,t_2) = \frac{1}{|t_1|} \int_{t_1} d_s(p,t_2) dp
\]

The Hausdorff distance from a trajectory point \(p\) to another trajectory \(t_2\) is defined as

\[
d(p,t_2) = \min_{q \in t_2} \{d(p,q)\}
\]

The distance between trajectories \(t_1\) and \(t_2\) is the maximum of their one way distances.

\[
\text{dist}(t_1,t_2) = \max \{\text{dist}_{ow}(t_1,t_2), \text{dist}_{ow}(t_2,t_1)\}
\]

Clearly the \(\text{dist}_{ow}(t_1,t_2)\) is not symmetric but \(\text{dist}(t_1,t_2)\) is symmetric. Note that \(\text{dist}_{ow}(t_1,t_2)\) is the integral of the shortest distances from points in \(t_1\) to \(t_2\).
A. Generalized Trajectory Cluster Routine

Trajectories are grouped into clusters using the threshold. Here the threshold is considered as a maximum value, such that all trajectories (belongs to each user) are grouped into a single cluster. The trajectory cluster routine contains the following stages:

1. Dissimilarity matrix for trajectories will be computed using the Hausdorff distance.
2. Using Initialization Algorithm (Table. 1) trajectories are grouped into initial clusters;
3. Using RepTraj Algorithm (Table. 1) representative trajectories are computed.
4. By considering the trajectories received from step 3, as initial cluster centres, using Re-cluster Algorithm (Table. 1) recompute clusters and their representative trajectories until there is no change in the representative trajectories.

| TABLE I Trajectory Clustering Algorithm |
|----------------------------------------|
| **Algorithm – 1 (Initialization)**     |
| a. Take first sample as first cluster. Classify all the remaining trajectories into this cluster if they are within the threshold. |
| b. Take a trajectory (sequentially) which is not already classified into any of the cluster and consider it as a new cluster. Take all the other trajectories which are not kept in any of the clusters and keep in this cluster if they satisfy the threshold limit. |
| c. Repeat step b till no new clusters are added. |
| **Algorithm – 2 (Representative Trajectory of Cluster C)** |
| For each Trajectory of cluster C calculate cumulative dissimilarity with all other trajectories of the same cluster C. Select the trajectory which is having minimum cumulative dissimilarity and take this as representative trajectory of that cluster. |
| **Algorithm – 3 (Re-compute)** |
| 1. For each Trajectory calculate dissimilarity with all the K representative trajectories and classify to the cluster for which dissimilarity is low (if it is within the threshold). |
| 2. Re-calculate representative trajectories using Algorithm – 2. |

III. EXPERIMENTAL WORK

Proposed algorithm is experimented on the dataset obtained from 100 computer science undergraduate and graduate students of the JNTUK\(^2\). The students are asked to enroll and authenticate during one month period of this study. The users are asked to enter their registration number as the login ID and their self chosen random length string as the password. From the obtained samples, for each user, a user feature is created in the following way:

1. Each participant is asked to type his/her password five times in three random sessions.
2. From each session, using the above trajectory clustering algorithm, a repTraj is generated, like this, three repTraj are generated from the three random sessions.
3. Average dissimilarity between the three repTraj is considered as the userThreshold, and by clustering the three repTraj (obtained from step 2) a userRepTraj is generated.
4. The userRepTraj and the userThreshold are considered and stored as the user feature.

After obtaining the user feature, the participants are asked to face the authentication phase for computing FRR and FAR. For authenticating the user, as usual, the user is asked to enter his login ID and password, from the entered data, a trajectory is generated and dissimilarity is computed with the userRepTraj, if the dissimilarity is within the usersThreshold, then the user is considered as the authorised user.

Following Fig. 2, shows the obtained visualizations from the tool. Fig. 2-(i) shows the normal (familiar user with the computer) user visualization, in Fig. 2-(i) a, b, c shows the three random sessions and the red highlighted trajectory is the repTraj of the session respectively, and highlighted one in Fig. 2-(i)d is the userRepTraj, which is obtained by clustering the three repTraj shown in (i)-a, b and c. Similarly, Fig. 2-(ii) shows the example case of a novice user (with the computer) where the FRR is observed.

---

5http://jntukakinada.edu.in; dataset is available: http://sites.google.com/site/munaga71.
In this paper, we have presented a novel trajectory clustering technique for authenticating the user using keystroke dynamics. We have demonstrated the effectiveness of our solution by testing the proposed technique on the data obtained from the 100 computer science undergraduate and graduate students. As discussed in section 3, the technique proved to be efficient in supporting the administrator for authenticating the valid user among other users, even though the password theft occurred. Moreover, for the typing biometrics system, it was observed that the Hausdorff dissimilarity measure can be adopted for getting effective results, i.e., as shown in section 2 the Hausdorff dissimilarity measure is more sensitive for small variations. As a limitation of the proposed technique, during experimentation phase, we tested the toll on a novice user, where we observed the False Rejection Rate.

Even though the continuous monitoring concept of key stroke dynamics is available, but it can not be applicable for all applications (e.g., accessing ATM, where only small amount of time/keys used to access the system), this static concept of key stroke dynamics will be useful. User authentication supported by Keystroke dynamics has many applications in the today’s electronic world, especially applications where secure data transfer is mandatory.

REFERENCES

[1] R. Gaines, W. Lisowski, S. Press, and S. Shapiro, “Authentication by keystroke timing: Some preliminary results,” Rand Report R-256-NSF, Rand Corp, 1980.
[2] R. Joyce and G. Gupta, “Identity Authentication Based on Keystroke Latencies.” Communications of the ACM, Vol. 33, No. 2, p168-176, 1990.
[3] F. Monrose and A. D. Rubin, “Authentication via Keystroke Dynamics,” Proc. of the Fourth ACM Conference on Computer and Communication Security, Zurich, Switzerland, 1997.
[4] F. Monrose, M. K. Reiter, and S. Wetzal, “Password hardening based on keystroke dynamics,” International Journal of Information Security, pp. 69-83, 2001.
[5] Cho S, Han C, Han D, Kim H. “Web-based keystroke dynamics identity verification using neural network,” J Organ Comput Electron Commerce 2000;10(4):295e307.
[6] K. S. Sung and S. Cho, “GA SVM wrapper ensemble for keystroke dynamics authentication,” Proc. of International Conference on Biometrics, Hong Kong, pp. 654-660, 2006.
[7] K. Revett, S. Magalhaes, and H. Santos, “Developing a keystroke dynamics based agent using rough sets,” Proc. of The 2005 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology Workshop on Rough Sets and Soft Computing in Intelligent Agents and Web Technology, Compiegne, France, pp. 56-61, 19-22 September, 2005.
[8] K. Revett, “On the use of multiple sequence alignment for user authentication via keystroke dynamics,” Proc. of International Conference on Global eSecurity, University of East London, pp. 112-120, 16-18 April, 2007.
[9] Cheng Soon Ong and Weng Kin Lai, “Enhanced Password Authentication through Typing Biometrics with the K-Means Clustering Algorithm,” Proc. Of Seventh International Symposium on Manufacturing with Applications (World Automation Congress), Maui, Hawaii, June 11-16, 2000.
[10] Edmond Lau, Xia Liu, Chen Xiao, and Xiao Yu, “Enhanced User Authentication Through Keystroke Biometrics”, 6.857: Computer and Network Security, Final Project Report, Massachusetts Institute of Technology, December 9, 2004.
[11] H. Munaga, L. Ieronutti, and L. Chittaro, “CAST - A Novel Trajectory Clustering and visualization tool for spatio temporal data,” In IHCI-2009: Proc. of the First International conference on Intelligent Human Computer Interaction, pages 169–175, Springer, India, January 2009.
[12] H. Munaga, J. V. R. Murthy, and N. B. Venkateswarlu. “A Novel Trajectory Clustering technique for selecting cluster heads in wireless sensor networks,” International Journal on Recent Trends in Engineering, 1:357–361, May 2009.
[13] R. Agrawal, C. Faloutsos, and A. Swami, “Efficient similarity search in sequence databases”, In Proc. of the Fourth International Conference on Foundations of Data Organization and Algorithms, Chicago, pg. 69–84, 1993.
[14] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. “Fast subsequence matching in time–series databases”, In Proc. of the ACM SIGMOD Conference on Management of Data, pg. 419–429, 1994.
[15] Z. Struzik and A. Sibes, “Measuring time series similarity through large singular features revealed with wavelet transformation”, In Proc. of the 10th International Workshop on Database and Expert Systems Appl., pp. 162–166, 1999.
[16] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani, “Mining the stock market: Which measure is best?”, In Proc. of International conference on Knowledge discovery and data mining, pp. 487–496, 2000.
[17] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition”, IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-26(1):43–49, Feb. 1978.
[18] D. Berndt and J. Clifford, “Using Dynamic Time Warping to Find Patterns in Time Series”. In Proc. of AAAI94 Workshop on KDD, 1994.
[19] E. Keogh and M. Pazzani, “Scaling up Dynamic Time Warping for data mining Applications”, In Proc. 6th International Conference on Knowledge Discovery and Data Mining, Boston, MA, 2000.
[20] S. Park, W. Chu, J. Yoon, and C. Hsu, “Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases”, In Proceedings of IEEE ICDE, pages 23–32, 2000.
[21] D. P. Huttenlocher and K. Kedem, “Computing the minimum Hausdorff distance for point sets under translation,” In SCG ’90: Proceedings of the sixth annual symposium on Computational geometry. New York, NY, USA: ACM, 1990, pp. 340–349.