A Comparison of Keystroke Dynamics Techniques for User Authentication

Tanapat Anusas-amornkul
King Mongkut's University of Technology North Bangkok
1518 Pracharat 1 Rd., Wongsawang,
Bangsue, Bangkok 10800, Thailand
tanapata@kmutnb.ac.th

Kasem Wangsuk
King Mongkut's University of Technology North Bangkok
1518 Pracharat 1 Rd., Wongsawang,
Bangsue, Bangkok 10800, Thailand
digital_hyper@hotmail.com

***Abstract*—Recently, in the Internet, one of the most security vulnerabilities is a weak password setting. There are several ways to make the password harder to guess by increasing the number of characters, password complexity, or changing the password more often. However, using only passwords for authentication may not be enough because passwords can be written down or exposed to others easily. Therefore, several researchers are solving this problem by adding keystroke dynamics to a username or a password to strengthen the authentication process. In this work, three keystroke dynamics techniques, i.e. statistics using confidence interval, k-means clustering, and trajectory dissimilarity, are implemented and compared with the same dataset. The performance metric is accuracy. In addition, pseudocodes for the techniques are also presented. From the experiment, the trajectory dissimilarity technique gives the best accuracy at 96% among others.**

Keywords—comparison; keystroke dynamics; authentication

I. INTRODUCTION

Keystroke dynamics is the patterns of rhythm and timing created when a person types. It is based on a principle of timing to press and release keys on a keyboard. The hypothesis is that a rhythm for each user's typing is unique and can be used for a user authentication token. Basic features of keystroke dynamics are as follows.

Key hold time refers to the time that starts pressing any key and holds the key until releasing that key.

Interkey time is the time to change from one key to another key, which may have a positive or negative value. If the value is positive, the key is released before the next key is pressed. If the value is negative, the next key is pressed before the previous key is completely released or two keys are pressed on the overlap.

Latency time is the time to press any key until pressing the next key. It also equals to the time to release any key until releasing the next key. Fig. 1 shows the basic keystroke dynamics features as described previously and the keystroke dynamics features can be used to create a unique user profile by feeding into a keystroke dynamics analysis (KDA) technique.

978-1-4673-7825-3/15/$31.00 ©2015 IEEE

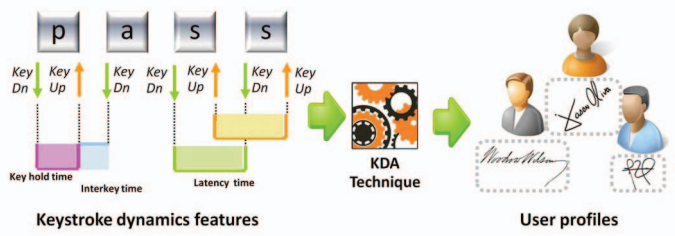

Fig. 1. Basic features for keystroke dynamics

Various techniques have been proposed to improve the performance of keystroke dynamics. Each technique measures the performance and calculates the accuracy with its own dataset. However, it is difficult to compare the accuracy with different dataset. Therefore, in this work, three techniques are carefully implemented and compared with the same dataset such that the comparison is fair enough to get a conclusion.

The first technique is a simple statistical method using only confidence interval [1]. The second technique is using k-mean clustering with learning windows [5]. The last technique is the trajectory dissimilarity technique [8]. The reasons for choosing these 3 techniques are as follows. The first technique is simple and uses technique similar to the third one. Another reason is that the second technique presents an interesting idea on how to adapt the user profile generation by using learning windows with high accuracy at 96.2%. The last technique is the trajectory dissimilarity technique, which is our own previous work and we need to compare with other works with the same dataset.

The paper is organized as follows. The next section is the related works, which review several related research in this area. Then, three selected keystroke dynamics techniques are described with pseudocodes. Next, the experiment information is explained with the data collection, performance metrics, and developed applications. In section 5, the results and discussions are presented and the last section is a conclusion.

II. RELATED WORKS

A keystroke dynamics analysis (KDA) is an interesting topic in the area of variability and instability of a user's typing rhythm. Many researchers have proposed several techniques to

make use of the unique keystroke rhythm for a user authentication.

For example, S. Haider, et al. [1] proposed several techniques using fuzzy logic, neural network, and basic statistics using confidence interval for user authentication. T. Limpanuparb [2] proposed advance statistical techniques using Counting of Abnormal Values (CAV) and Probability of Mean Absolute of Standard score (PMAS) approaches. R. Montalvao, et al. [3] proposed a single memoryless non-linear mapping of time intervals using histograms. We briefly summarized several techniques of related works as shown in Table 1 with the accuracy for each technique.

Several techniques were proposed for KDA, but each technique has pros and cons. For example, a basic statistical technique [1] is simple and takes less memory, including small space usage for storing data on a user profile. However, a weakness is that the accuracy of the authentication is low.

A combination of statistical techniques [2], CAV and PMAS, can improve the performance by using two threshold parameters which gave a high accuracy to 94.76%. Parameters used for the calculation were difficult to find appropriate values by using an exhaustive search, which was time consuming if the number of users was large.

TABLE I. KEYSTROKE DYNAMICS TECHNIQUES

Authors	Techniques	Accuracy (%)
S. Haider, et al. [1]	Neural, Fuzzy, Statistics using CI	92.50*
T. Limpanuparb [2]	Advance Statistics using CAV, PMAS	94.76
R. Montalvao, et al. [3]	Histogram Equalization	87.30
D. Tran, et al. [4]	Markov and Fuzzy	91.40
P. Kang, et al. [5]	K-means clustering, learning windows	96.20
R. Giot, et al. [6]	Support Vector Machine	93.04
C. Jiang, et al. [7]	Hidden Markov Models	97.46
K. Wangsuk, and T. Anusas-amornkul [8]	Trajectory Dissimilarity	96.00

Note: * not explicitly specified in the paper.

Another work is using k-means clustering [5] which reported the high accuracy. Data collected from 21 subjects were classified using k-means clustering integrated with moving windows and growing windows. The advantage of this idea was to maintain the performance of the system over time. Such method gave the performance of the system with the accuracy up to 96.2%. However, the concept of learning principles by using windows integrated with a classification technique brings high space requirements to store the pattern for each user. This weakness must be considered when a system with a large number of users is practically implemented. A trajectory dissimilarity technique [8] were proposed and a master trajectory profile is created to verify a user using Euclidean distances. The authors claimed to give the accuracy at 96%. Keystroke dynamics techniques were surveyed in [9] and presented the recently proposed works in this area.

All proposed works use their own dataset and it is difficult to compare with others. Therefore in this work, three KDA techniques are implemented and compared with the same dataset.

III. KEYSTROKE DYNAMICS TECHNIQUES

In this section, three keystroke dynamics techniques are described in details along with pseudocodes for implementation. Each technique is presented with some parameters related to its algorithm.

A. Statistics using confidence interval

This technique is simple by using a basic statistics, which is a confidence interval as a pass or fail authentication [1]. The authors used only one KDA feature, which is the interkey time to verify a user. A user passes the authentication process if the interkey time is in the confidence interval range by calculating from equation 1.

$$x_i \pm z\,\sigma \tag{1}$$

Where x_i is an average interkey time for key i, z is a standard normal distribution value, and σ is a standard normal deviation value. The pseudocode for this technique is shown in Fig.2.

```
Algorithm StatisticsWithCIAuthentication
  users[] ← All User Data, Allow Fail Key ← Input Allow Fail Key, z ← 0;
  Repeat
    For i ← 1 to 20 do
      Interkey Confidential ← Calculate Confidential Window (users [i-1], z )
      P1 ← 0, F1← 0, P2← 0, F2← 0
      For j ← 1 to 30 do
        Num of Fail Key ← VerifyKey (users[i-1].RealUserTrajectory[j-1], z )
        If (Num of Fail Key < Allow Fail Key)
          P1 = P1+1
        Else
          F1 = F1+1
        EndIf
        Num of Fail Key2← VerifyKey (Users[i-1].ImposterTrajectory[j-1], z )
        If (Num of Fail Key2 < Allow Fail Key)
          P2 = P2+1
        Else
          F2 = F2+1
        EndIf
  FAR ← F1/30, FRR ← P2/30
  Until FAR = FRR or FAR nearest to FRR
  Return ERR ← Max (FAR, FRR)
End
```

Fig. 2. Pseudocode for the statistical technique using confidence interval

B. K-means clustering

The k-means clustering technique was proposed in [5] by using a k-means algorithm based on Euclidean distance as the authentication classifier. An interesting idea for this paper is that a learning window is used for continually retraining the classifier over time because the authors gave the assumption that the keystroke dynamics can be changed when a user is familiar with a password typing.

Two types of learning windows, i.e. moving and growing windows, are proposed to continual update a user profile. For a moving window technique, the number of data used for generating a user profile is the same but the most recent typing data is added to create a profile, while the oldest data is removed. For a growing window technique, the number of data is increasing over time to create a user profile.

In this technique, the number of cluster is set to 3. Two KDA features, i.e. interkey time and hold time, are used for a user classification and the pseudocode, presented in the original paper, is shown in Fig.3.

Algorithm KMeansClusteringKeystrokeAuthentication
Step 1: Perform K-Means clustering with the training patterns
$[C_1,...,C_K] = K - Means(X, K)$
(C_K: the members belonging to K-th cluster)
(X: training patterns, K: the number of clusters)

Step 2: Find the closest cluster prototype of the test pattern y_i
$k = \arg_{i \in 1,..,K} \min dist(y_i, P_i)$
(P_i: the prototype of the cluster C_i)

Step 3: Authentication
If $dist(y_j, P_i) < M \times \frac{1}{|N_k|} \sum_{x_i \in c_k} dist(x_i, P_k)$
(N_k: the number of patterns in k-th cluster, M : Threshold coefficient)
Grant access(y_j is considered as a valid user's typing pattern)
Else
Deny access(y_j is considered as an impostor's typing pattern)
EndIf
End

Fig. 3. Pseudocode for the k-means clustering technique

```
Algorithm TrajectoryDissimilarityAuthentication
  users[] ← All User Data
  level ← 0
  Repeat
    For i ← 1 to 20 do
      allowance ← Calculate Allowance (users [i-1], level +0.01)
      P1 ← 0, F1← 0, P2← 0, F2← 0
      For j ← 1 to 30 do
          dissimilarity← CalcDissimilarity (
                  users[i-1].RealUserTrajectory[j-1])
          If (dissimilarity < allowance )
            P1 = P1+1
          Else
            F1 = F1+1
          EndIf
          dissimilarity2←CalcDissimilarity (
                  users[i-1].ImposterTrajectory[j-1])
          If (dissimilarity2 < allowance )
            P2 = P2+1
          Else
            F2 = F2+1
          EndIf
    FAR ← F1/30, FRR ← P2/30
  Until FAR = FRR or FAR nearest to FRR
  ERR ←  Max (FAR, FRR)
  Return level and ERR
End
```

Fig. 4. Pseudocode for the trajectory dissimilarity technique

C. Trajectory dissimilarity

The last technique is the trajectory dissimilarity technique [8]. A trajectory graph is proposed to create a user profile and the classification technique is to use a simple *allowance* parameter as shown in equation 2.

$$allowance = 0.1646 (level * 0.0381) \quad (2)$$

In this technique, selected KDA features are the interkey time and latency time, used for creating a trajectory profile. A Euclidean distance is used for finding dissimilarity between a master trajectory profile and a current trajectory data. A pseudocode for this technique is shown in Fig.4.

IV. Experiment

In this work, a username data is collected instead of password. This is because a username is rarely changed and users are familiar with username typing rhythms and it is a good data to create unique biometric data. This section is divided into three subsections, which are data collection, performance metrics, and implementations.

A. Data collection

Each subject was assigned to type in a username in 3 sets. Each set was collected at least one day apart to eliminate the variation over time and to be consistent with the use in a real system. For each set, a user typed in a username 10 times, and each time was delayed for at least 5 seconds apart to reduce the variation of behavior while typing. The format of a username is firstname followed by the first character of lastname.

For example, if a subject's name is "**Kasem Wangsuk**", a username will be "**kasemw**". In addition, the assumption that the user is familiar with typing brings stability and uniqueness to the data collection. In summary, each subject was assigned to type 3 sets of username, each set contains 10 times of username typing data. Therefore, the amount of individual data was 30 records per user. Only the first set of data will be used to create a master profile for each subject and the remaining two data sets were used to measure the performance. Three subjects were assigned to act as imposters. Each imposter requires to type one set of username of all subjects. It means that each subject has 3 sets or 30 records of forged username typing data. After collecting data of all 20 subjects, we have collected all the data up to 1,200 records of username typing data.

B. Performance Metrics

The ability of a KDA technique to correctly differentiate a genuine user and an imposter is the effectiveness of such technique. Performance metrics are described as follow.

False Rejection Rate (FRR) is the percentage ratio between falsely denied genuine users against the total number of genuine users accessing the system.

$$FRR = \frac{Total\ number\ of\ falsely\ denied\ genuine\ users}{Total\ number\ of\ genuine\ users} * 100 \quad (3)$$

False Acceptance Rate (FAR) is the percentage ratio between falsely accepted imposters against the total number of imposters accessing the system. A smaller FAR indicates less imposter accepted.

$$FAR = \frac{Total\ number\ of\ falsely\ accepted\ imposters}{Total\ number\ of\ imposters} * 100 \quad (4)$$

Equal Error Rate (EER) is a parameter where False Rejection Rate (FRR) and False Acceptance Rate (FAR) are equal. Smaller EER is better. It is a common measure of the performance of the KDA.

Accuracy is related to EER since the lower the EER, the higher the accuracy. It is shown in equation 5.

$$Accuracy = 100 - EER \quad (5)$$

This paper uses the accuracy as a main performance metric to compare the three techniques.

C. Implementations

Applications for each technique were developed using C# programming language. The user interface (UI) for each technique is shown in Fig.5 to Fig.7, respectively.

Fig. 5 shows the application user interface for a statistical technique using confidence interval. The dotted lines on the upper graph show the allowable gap for each key. A genuine user should type each character within this time interval.

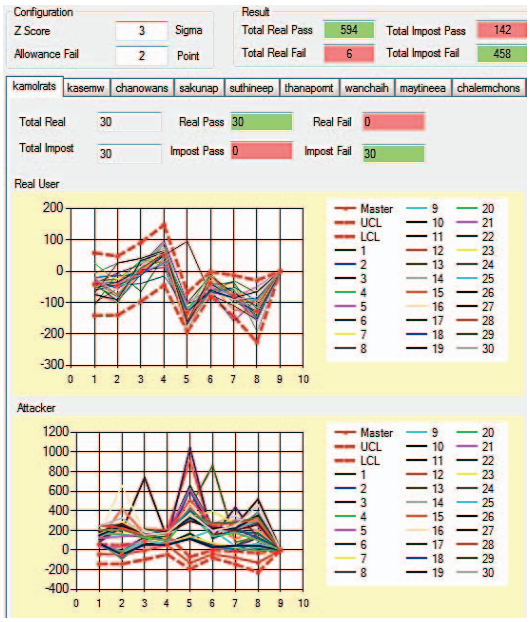

Fig. 5. A developed statistics using CI application

Fig. 6 shows the application user interface for the k-means clustering technique. There are 3 clusters, showing in red, green and blue colors with 3 centroids. Both moving and growing windows were implemented.

Fig. 7 shows the application user interface for the trajectory dissimilarity technique. The red line in the graph is a master trajectory profile, which is used to compare with another trajectory data from a user.

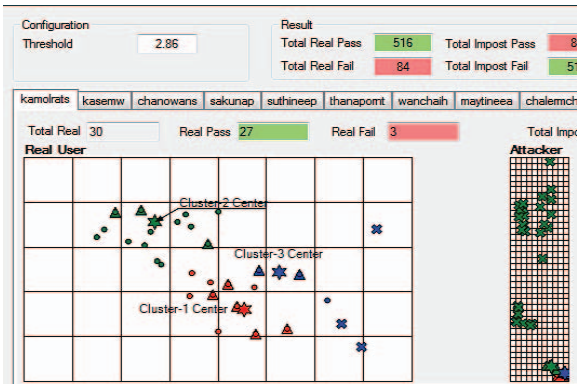

Fig. 6. A developed k-means clustering application

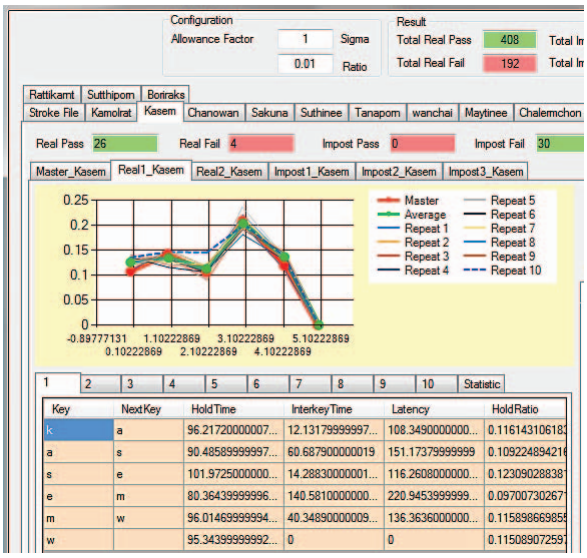

Fig. 7. A developed trajectory dissimilarity application

V. Results and Discussions

From our experiments, the dataset is the same to measure the performance of each technique. Table 2 shows the summary of the KDA technique comparisons. There are differences in the original EER and tested EER. The statistical technique using confidence interval performs better in our experiment, but the k-means clustering technique performs worse. Since the dataset is the same as our previous work, the trajectory dissimilarity technique gives the same EER.

TABLE II. Keystroke Dynamics Technique Comparisons

KDA Techniques	Original EER(%)	Tested EER(%)	Accuracy (%)	Features
Statistics using confidence interval	7.5	5.58	94.42	Interkey time
K-means clustering	3.8	12.25	87.75	hold time + interkey time
Trajectory dissimilarity	4.0	4.0	96.0	Interkey + latency time

The parameters for each technique are optimized to find the best EER and the results show that the trajectory dissimilarity technique gives the best EER and accuracy percentage among others in this dataset. This is because the dataset was originally collected for the trajectory dissimilarity technique. In addition, the source codes for other two techniques are not available such that the implementations and parameter settings may not be the same as the original works. However, we try our best to replicate the algorithms with the best results. Another difference is that the previous two techniques use keystroke dynamics for passwords but the trajectory dissimilarity technique uses keystroke dynamics for usernames. However, there are no differences in the data collections for each feature.

VI. Conclusion

Keystroke dynamics is an interesting research area for strengthen a username and password authentication scheme, which is widely used today. For each work, researchers collected their own dataset and measured the accuracy. However, it is not fair to compare the results with other works even with the same performance metrics. Therefore, the objective of this paper is to compare different KDA techniques with the same dataset in order to justify the performance for the three techniques.

In this paper, we presented the pseudocodes for three KDA techniques, i.e. statistical technique using confidence interval, k-means clustering and trajectory dissimilarity techniques. Then, the applications used for testing each technique were implemented using C# language. The results show that the trajectory dissimilarity technique gives the best accuracy at 96%. Therefore, the experiment is verified that, with this dataset, this technique gives potentially good performance and it can be used for an additional secret to strengthen the authentication scheme. It is important to note that the KDA techniques are keyboard dependent. If a user changes a keyboard, a user profile have to be recreated.

Our future work is to apply the trajectory dissimilarity technique to defend against an account lockout attack, which is the attack that an attacker attempts to lockout user accounts by intentionally inputting wrong password several times. The result is that a real user cannot access his account for several minutes depending on an account lockout policy setting. This attack is easy to deploy and a user who is in a hurry to access his important account can be greatly affected by this attack.

References

[1] S. Haider, A. Abbas, and A. K. Zaidi, "A multi-technique approach for user identification through keystroke dynamics," *IEEE International Conference on Systems, Man and Cybernetics*, pp.1336-1341, 2000.

[2] T. Limpanuparb, "The Enhancement of Password Security System Using Key Stroke Verification," *NECTEC Technical Journal,* vol. 4, 2004.

[3] J. Montalvao, C. A. S. Almeida, and E. O. Freire, "Equalization of keystroke timing histograms for improved identification performance," *Telecommunications Symposium, 2006 International*, pp.560-565, 2006.

[4] D. Tran, W. Ma, G. Chetty, and D. Sharma, "Fuzzy and markov models for keystroke biometrics authentication," in *The 7th WSEAS International Conference on Simulation, Modeling and Optimization, SMO 2007*, pp.89–94, Stevens Point, Wisconsin, USA, 2007.

[5] P. Kang, S.-s Hwang, and S. Cho, "Continual Retrain of Keystroke Dynamics Based Authenticator," *Springer-Verlag Berlin Heidelberg*, pp.1203-1211, 2007

[6] R. Giot, M. EI-Abed, and C. Rosenberger, "Keystroke Dynamics with low constraints svm based passphrase enrollment," in *The 3rd IEEE international conference on Bio-metrics: Theory, applications and systems, BTAS 2009*, pp.425–430, Piscataway, NJ, USA, 2009

[7] C.-H. Jiang, S. Shieh, and J.-C. Liu, "Keystroke Statistical learning model for web authentication," in *The 2nd ACM symposium on Information, computer and communications security, ASIACCS 2007*, pp.359–361, New York, NY, USA, 2007.

[8] K. Wangsuk and T. Anusas-amornkul, "Trajectory Mining for Keystroke Dynamics Authentication," *Procedia Computer Science,* vol. 24, pp. 175-183, 2013.

[9] P. -S. Teh, A. B. J. Teoh, and S. Yue, "A Survey of Keystroke Dynamics Biometrics," *The Scientific World Journal*, vol. 2013, 2013.