Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 4, 2020

Review Article

REAL – TIME BIOMETRIC AUTHENTICATION BASED ON KEY STROKE DYNAMICS

Alaa B. Baban, Ahmed M. Alkababji, Mustafa S. Oassab

Computer Science Dept, College of Science and Engineering, Bayan University, Kurdistan, Erbil, Iraq
Computer Engineering Dept, College of Engineering, Mosul University, Mosul, Iraq
Computer Engineering Dept, College of Engineering, Mosul University, Mosul, Iraq

Received: 13.12.2019 ***Revised: 02.01.2020*** ***Accepted: 03.02.2020***

Abstract
In this paper we discuss the ability to implement keystroke dynamics in a simple way by taking few, but effective, features and run the authentication session in a real-time manner. MATLAB 2019 is used to implement the keystroke biometrics and 15 people were included in the test aging from 20 to 30 years, and 10 samples are taken from each volunteer to train the Back Propagation Neural Network (BBNN) in order to classify the various entered patterns according to the obtained features. The results are satisfying were the FRR has a value of 27% and FAR is about 18%. This approach, named Simple but Effective (SbE), can be done using non-convenient features which in turn makes it unique, in addition to its simplicity that makes it relatively consumes less resources than previous experiments.

Keywords—biometrics, keystroke dynamics, feature extraction, classifiers, neural network, real-time application.

 DOI: http://dx.doi.org/10.31838/jcr.07.04.96

INTRODUCTION

With the ever increasing demand for more secure access control in many of today's security applications, traditional methods such as PINs, tokens, or passwords fail to keep up with the challenges presented because they can be lost or stolen, which compromises the system security.

On the other hand, biometrics based on "who" is the person or "how" the person behaves present a significant security advancement to meet these new challenges [1].

Where compromised passwords and shared accounts are frequently exploited by both external attackers and insiders, if we had some means, other than knowledge of a password, with which to identify exactly who is logging into an account, and to discriminate between the *genuine user* of an account and an *impostor*, we could significantly curb these security threats [2].

Here comes the significance of Keystroke Dynamics (KD), which is an effortless behavior-based method for authenticating users, that employs the person's typing patterns for validating his/her identity. Keystroke dynamics is "not what you type, but how you type." [3].

Compared to other biometrics, keystroke biometrics has additional desirable properties due to its user-friendliness and non-intrusiveness. Keystroke dynamics data can be collected without a user's cooperation or even awareness [1].

In this approach, the user types in text, as usual, without any extra work to be done for authentication. Moreover, it only involves the user's own keyboard and no other external hardware. These criteria make keystroke dynamics an excellent alternative or add on to the conventional ID/password authentication scheme [3].

Continuous authentication is possible using keystroke dynamics just as a mere consequence of people's use of computers. Unlike many other biometrics, the temporal information of keystrokes can be collected to ascertain a user using only software and no additional hardware. In summary, keystroke dynamics biometrics enables a cost effective, user friendly, and continuous user authentication with potential for high accuracy. In addition, keystroke biometrics can use "static text", where keystroke dynamics of a specific pre-enrolled text, such as a password, is analyzed at a certain time, e.g., during the log on process, while for more secure applications, "free text" should be used to continuously authenticate a user [1].

Keystroke dynamics verification is based on how a user types at a terminal equipped with a keyboard, which may belong to a personal computer, or be a generic interface equipped with keys which can be pressed. With respect to biometric modalities such as fingerprints, or iris, keystroke dynamics allows performing recognition on mobile devices without requiring any additional dedicated hardware. Moreover, it may require limited storage and computational resources, and its users' acceptability is very high [4].

LITERATURE REVIEW

Keystroke biometrics was first investigated in 1977 whether users could be distinguished by the way they type their names, [2]. As early as 1980, researchers have been studying the use of habitual patterns in a user's typing behavior for identification [5].

In early 1980's, a preliminary study on keystroke dynamics based authentication using the T-test on digraph features was held. Later, the extraction of keystroke features was done using the mean and variance of digraphs and trigraphs. The report of using the Euclidean distance metric with Bayesian-like classifiers shows that correct identification rate of 92% for their dataset containing 63 users [1].

Studies showed that there is a significant variability with which typists produce each digraph, and hence the use of pooled estimate digraph latency variability is inappropriate.

An additional limitation of the digraph latency based technique is the use of a single low-pass temporal filter for all typists for the removal of outliers. The rationale for this approach is that digraphs with abnormally long latencies are not likely to be representative of the authorized user's typing. While this seems like a reasonable assumption it has recently been shown that one filter value for all typists does not yield optimal performance [5].

Furthermore, the median inter-key latency of expert typists is approximately 96 ms, while that of novice typists is near 825 ms. Therefore, the 500 ms low-pass filter used excludes many keystrokes typical of novice typists, while at the same time, includes many keystrokes which are not representative of an expert typist. Other studies showed that the use of digraph-specific measures of variability instead of one low-pass filter can lead to measurable improvements in verification accuracy. Moreover, the approach of keystroke verification that uses the key down-to-down time as the base unit of measure, but this measure may be further delineated into two orthogonal

components; the total time the first key is depressed (i.e. keystroke duration), and the time between a key is released and the next key is pressed (i.e. keystroke latency). Substantially improved performance results based on using the bivariate measure of latency with an appropriate distance measure [5].

Then there was a proposal to use the relative order of duration times for different n-graphs to extract keystroke features that was found to be more robust to the intra-class variations than absolute timing. It was demonstrated that the new relative feature, when combined with features using absolute timing, improved the authentication performance using free text [1].

Some neural network approaches have been undertaken in the last few years. While the back-propagation models used yield favorable performance results on small databases, neural networks have a fundamental limitation in that each time a new user is introduced into the database, the network must be retrained. For applications such as access control, the training requirements are prohibitively expensive and time-consuming. Furthermore, in situations where there is a higher turnover of users, the down time associated with retraining can be significant [5].

Recently, keystroke dynamics has become an active research area due to the increasing importance of cyber security and computer or network access control. Most of the existing approaches focus on static verification, where a user types specific pre-enrolled string, e.g., a password during a login process, and then their keystroke features are analyzed for authentication purposes. Only a few research studies address the more challenging problem of keystroke biometrics using "free text", where the users can type arbitrary text as input [1].
The four most generally considered keystroke dynamic features are illustrated as in figure 1, which are defined by keystroke times of button presses and button releases, i.e., PR (press–release), PP (press– press), RP (release–press) and RR (release–release) times. More precisely, for example, the PR time is the time interval between pressing and releasing of the same key [6].

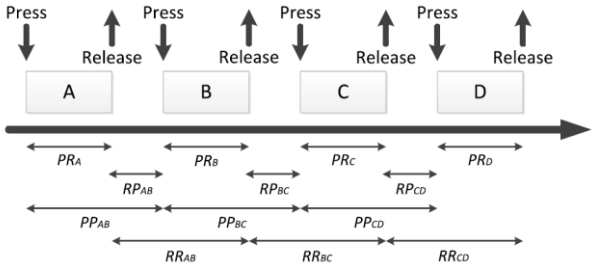

Figure 1: The four most generally considered KD features.

In addition to these four keystroke dynamic features, other keystroke biometrics had been proposed. The ''pause rhythm'' method was proposed and required users to add some pauses when entering their passwords. However, since one had to remember many extra information for the ''new'' password, such as where and the number of pauses inserted, and time length of each pause, this method was not only to conflict with the aim that the biometric features should be collected naturally, but also increase users' input burden, so it was not been widely adopted [6].

Proposed Approach

In this section, the proposed keystroke-based recognition system is illustrated. The approach relies on the analysis of keystroke dynamics referred to static text input using a normal laptop keyboard. To make it as simple as possible, only 4 features are extracted, two of them are considered as main features and the rest are supportive features.

ENROLLMENT

In the enrollment phase, the user is asked to enter a fixed text of 15 characters, and should go through the process of entering the fixed text for 15 times distributed on a three sessions, each session is typing the text for 5 times.

The reason behind making short sessions is for user's self-comfort, as collecting such data can be physiologically annoying if the session lasts for a long time. When a user enters a session of 30 to 60 seconds, he/she can stay focused and then take a rest before moving to the next session, this enhances the data collection as the user is becoming ready and not being tired of typing.

As the proposed approach is for login time only, and does not continue checking in the background, the user is expected to be relaxed and the typing rhythm is almost the same.
The target is to enroll a total number of 12 people; volunteers are aging from 30 to 50 years and including both genders. A sample of the collected data is shown in figure 2.

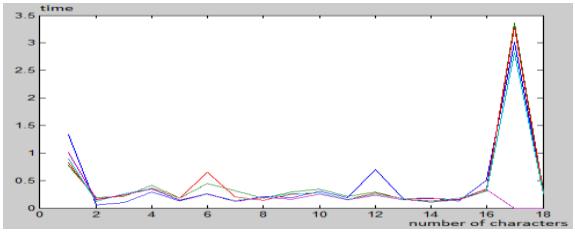

Figure 2: A sample of the collected data.

FEATURE EXTRACTION

As mentioned earlier, a total of four features are adopted in the proposed approach, the main two features are shown in figure 3. Where the first feature takes the timestamp between two presses (press latency) and the second feature is the total latency of typing the fixed text. These two features are taken as main features to be used in distinguishing different users.

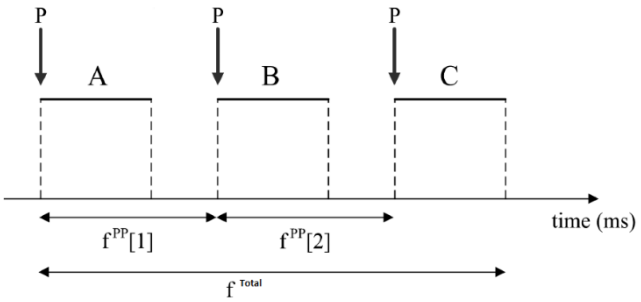

Figure 3: The main KD extracted features.

Other supportive features are considered to make the system more robust which are; an Enter key that should be pressed in the beginning of the typing, the other is to press a left-click key at the end of typing. These two extra keys are added not only to specify the start and end of the typing process, but also to use it as an indirect way of collecting more data from the user that can help in the authentication step in the real test of the proposed system.

CLASSIFICATION

After the creation of the data set, it should be used to classify the users through the use of one of the classification methods. Standard Neural Network (NN) and Back Propagation Neural Network (BPNN) are to be used and compared in order to achieve the best results.

AUTHENTICATION

After creating the database of different users, now the system is ready to be tested. When a user comes to login, the same fixed text in the enrollment phase should be entered again. Once there is no typo error, the extracted features are compared with the one in the database and the decision is made accordingly.

The proposed system is shown in figure 4.

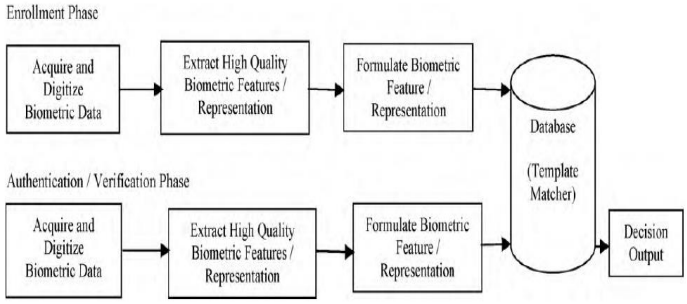

Figure 4: The proposed KD system.

Experiment

After suggesting the system approach, the system now can be implemented. The tools that are used include; a SONY VAIO laptop model PCG-71614M with Windows 7 Ultimate as an operating system, in addition to MATLAB 2013a software to design and implement the KD system.

After gathering the dataset and train the network, the database is now ready to be used. In order to login, the user will be asked to press enter first, then type the 15 character fixed text and press left-click when finished. By doing this, the features will be extracted and compared to the database and then the decision is made to be either a genuine user or an impostor.

Two error rates were used to infer the system performance, namely: False Accept Rate (FAR) and False Reject Rate (FRR). FAR is the percentage of impostors who have successfully gained access to the system whereas FRR indicates the percentage of legitimate users who were denied access to the system. Equations (1) and (2) are used to compute FAR and FRR, respectively.

$$\text{FAR} = \frac{\text{Number of false accept attempts (accepted imposters)}}{\text{Total number of impostor attempts}} = \frac{\textit{False Positive}}{\textit{True Negative} + \textit{False Negative}} \quad (1)$$

$$\text{FFR} = \frac{\text{Number of false reject attempts (rejected legitimate users)}}{\text{Total number of legitimate attempts}} = \frac{\textit{False Negative}}{\textit{True Positive} + \textit{False Positive}} \quad (2)$$

RESULTS AND ANALYSIS

Several experimental tests have been conducted to analyze the proposed keystroke dynamics-based verification system. The system is kept simple in order to achieve best real-time performance.

CLASSIFICATION USING BACK PROPAGATION (BPNN)

The adopted classification method is the BBNN, which contain 3 hidden layers and 10 nodes in the first layer and other hidden layers, and 8 nodes in the output layer, the used training function is TRAINCGP with adaptation linear function TRAINGDM, and the transfer function is logsig for the collected dataset, achieving the best mean square error of 10^{-8}, the rest of the dataset is used in the testing phase to test both legitimate users and imposters, so the security of the system can be tested, as:

7 legitimate users to be trained.
2 imposters to be trained.
3 imposters not to be trained.

Having two types of imposters, internal and external imposters, makes the system more robust to both types of attackers.

Each person gives 15 samples and only 10 of which are used for training and the rest for testing purpose. The results are shown in table 1 below:

Table 1: results of using BPNN in KD experiment.

Ratio	BPNN
Error	10^{-8}
FAR	18%
FRR	27%

DISCUSSION AND FUTURE WORK

As noticed in the results, the FAR is kept as low as possible to avoid imposters and the FRR is acceptable due to the wide variation in behavioral biometrics that can be changed according to the person's heath, tiredness, mood and many other factors.

The results can be enhanced by many means for instance; by collecting more samples for each person, collecting samples in different situations, positions and times, increasing the number of people including both legitimates and imposters, and finally trying new neural networks, layers, number of nodes and functions, these can be considered as future work.

CONCLUSION

In conclusion, KD is an easy to implement, transparent to user and costless biometric that can be effective when combined with a password to login in the case of fixed-text, or can continually monitor the keystrokes to prevent imposters from accessing the already logged-in page by running it in the background as a free-text KD biometrics.

Our experiment proved that even with a small number of samples provided by people and using the well-known, easy-to-handle neural networks, the KD system can still be created and gives acceptable results.

ACKNOWLEDGMENT

The authors wish to extend their gratitude to the participants who were involved in this experiment for the time they took out of their busy schedules to contribute in this study.

REFERENCES

1. Y. Zhong, Y. Deng and A. K. Jain, "Keystroke dynamics for user authentication," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Providence, RI, 2012, pp. 117-123.
2. K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, Lisbon, 2009, pp. 125-134.
3. Alsultan, et al., "Improving the performance of free-text keystroke dynamics authentication by fusion," Applied Soft Computing, 2018, vol. 70, pp. 1024-1033.
4. Emanuele Maiorana et al. , "Keystroke Dynamics Authentication for Mobile Phones, *" Proceedings of the 2011 ACM Symposium on Applied Computing (SAC),* TaiChung, Taiwan, 2011, pp. 21-26.
5. Fabian Monrose, Aviel D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, 2000, vol 16, issue 4, pp. 351–359.
6. Cheng-Jung Tsai, Kuen-Jhe Shih, "Mining a new biometrics to improve the accuracy of keystroke dynamics-based authentication system on free-text," *Applied Soft Computing*, 2019, vol 80, pp. 125-137.