

Keystroke Rhythm and Intensity as Biometrics for User ID

Lucas Hansen and Lindsay Willmore

Project Checkpoint – CS229

December 14, 2013

Abstract *Keystroke dynamics is a biometric measure of an individual's typing pattern, with applications in user authentication and password hardening. Past work has explored identifying users based upon detailed keystroke timing information, including the delay between and duration of each tap of a key. Here we explore an additional factor: keystroke intensity. Without imposing the need for additional infrastructure (e.g. pressure-sensitive keyboards), we use the laptop's built-in microphone to obtain an audio recording of the user's typing. Using timing information, we extract the volume intensity at the time of each keystroke. Our final algorithm uses both keystroke latency and intensity as features to distinguish between valid users and impostors. We are able to show that the additional intensity information greatly improves user classification by achieving final FAR and FRR of below 1% and 25% respectively.*

1 Introduction

We will focus on the recognition of a particular user out of a group of "imposters", based on keystroke dynamics information gathered while the user or imposter is typing a fixed password. In order to model this situation, we collected training examples of the legitimate user typing in the password, as well as a large number of examples of "imposter" password attempts. Given this data we built a model which is able to reliably distinguish between login attempts by the user and by an "imposter". The rate at which our model incorrectly identifies an imposter as the legitimate user is defined as the False Acceptance Rate (FAR), and the rate at which the valid user is rejected is known as the False Rejection Rate (FRR).

Our algorithm uses both the timing between keypresses and the physical intensity of each stroke. This particular method of biometric identification is attractive because of the low resource requirements that it imposes. While more robust authentication schemes exist (e.g. retinal or fingerprint scanning), they require significant infrastructure. Biometric identification based off of keystroke dynamics requires nothing that is not already built into a normal personal computer. For this reason, over the past 15 years there has been a lot of published work on the use of keystroke dynamics for authentication and enhanced security. Nearly all of this work has focused on data that can be harvested from the keyboard. However, we propose using an equally elegant and

convenient method to collect additional meaningful biometric data, which will improve authentication accuracy and robustness.

Almost every modern laptop includes a built-in microphone. This microphone is fixed in place, easily accessible, and located very close to the laptop's keyboard. These properties make it ideal for measuring the intensity with which a key is pressed. We extend on existing authentication techniques based off keystroke dynamics by using the intensity information harvested from the microphone.

2 Procedure

2.1 Data Collection

We developed a simple MATLAB program to extract data from a user typing a password of a set length. The experiment is set up in a semi-controlled environment, where the user sits alone in a quiet room and proceeds to type the password "andrew ng" 100 times into one of two laptops of the same make and model. There is a small gap in time between each password entry. The user is given a visual signal to begin typing the password, but the typed password characters do not appear on the screen. These precautions were taken to minimize distractions and optimize consistency between typing trials; however, we do recognize that severely restricting the environment of the user reduces the generalizability of our conclusion.

We begin our audio recording just prior to begin-

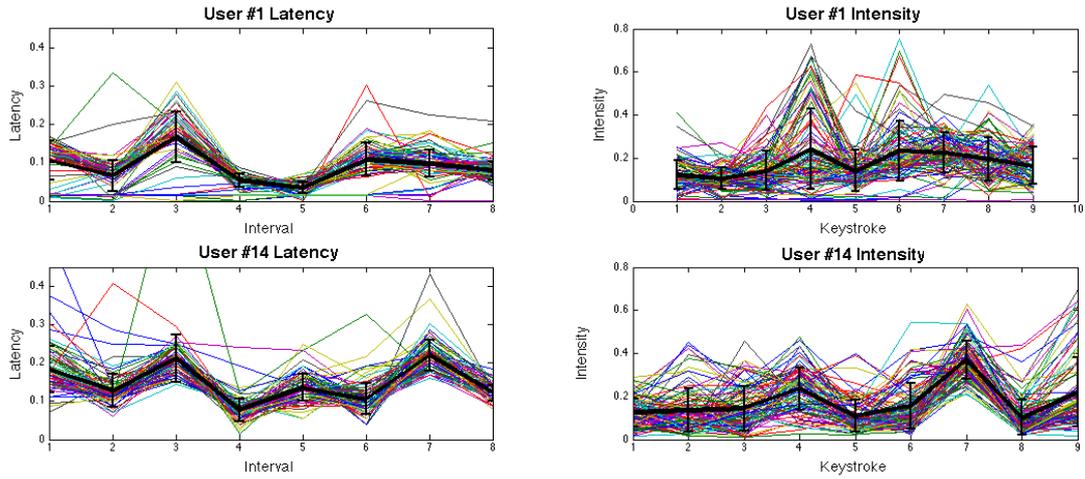


Figure 2.1: Plots of raw latency and intensity data values for 2 users, displaying distinct keystroke patterns of different users (Top: User 1, Bottom: User 14)

ning of the first keypress of each separate password trial. Then we record the latencies between keypresses, which indicate the times in the audio data where peak amplitudes should be. The volume spike closest to the time of the keypress is taken as a measure of keystroke force.

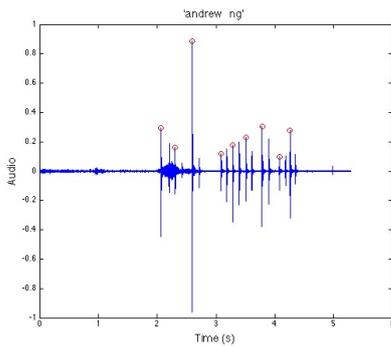


Figure 2.2: Intensity extraction: audio recorded during duration of typing, red points indicate identified keystroke intensities

2.2 Preprocessing, Normalization, and Visualization

The data that we collected required relatively little processing. The primary preprocessing step is the extraction of the keystroke intensity estimate from the audio data collected by each trial.

After extracting the intensity information from the audio data generated by each trial all of our data is entirely digital. In the process of recording, we also maintained a record of the characters typed. After recording a user’s 100 trials, we removed trials with misspellings (i.e. the character sequence typed did not match 'andrew ng'). This resulted in a final data set of 1317 total examples taken from

14 users, where each user had anywhere from 79 to 99 successful trials. This data was composed of 17 features - 8 latencies and 9 intensities from each of the 9 characters in the password.

Some users’ intensities were recorded using an earlier version of our data extraction protocol, resulting in absolute values approximately 10 times lower than those of users sampled using the updated extraction protocol. We therefore multiplied those low intensities by a correction factor of 10.

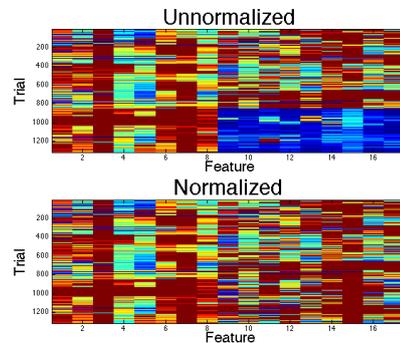


Figure 2.4: Heat maps of full data set before and after normalization, consecutive trials grouped by user.

Figure 2.4 reveals the distinct typing patterns of each user represented by a solid band of consecutive password trials. Of note are those features which show little variation, such as feature 3 - representing the latency between 'd' and 'r', and those features which vary significantly between users, such as feature 5 - representing the latency between 'e' and 'w'.

To confirm our features’ ability to distinguish between individual users we performed Principal Component Analysis (PCA) and plotted the first

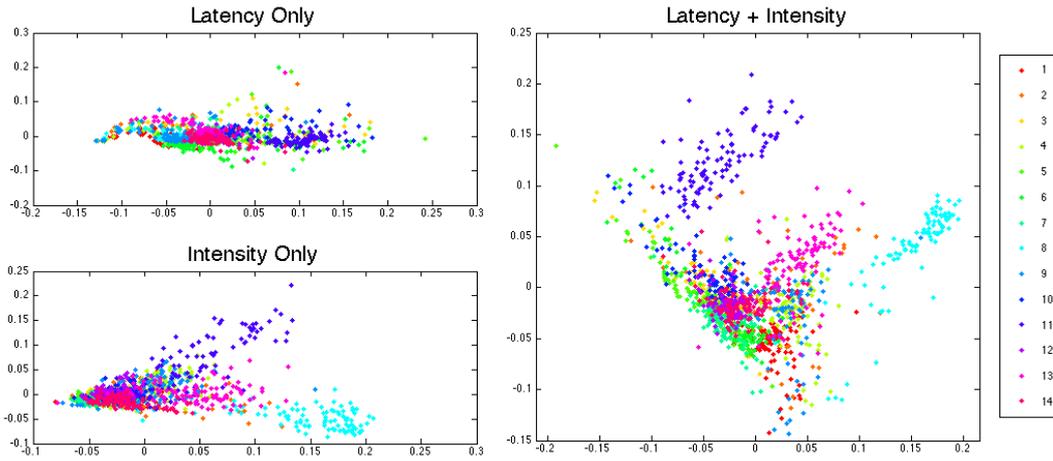


Figure 2.3: PCA: using subset or full set of keystroke features, colors representing distinct users

principal component against the second. For this visualization only, we performed additional normalization on the data, ensuring that each feature had zero mean and unit variance. This normalization drastically enhanced the explanatory power of our PCA plots.

We performed this procedure using our full set of features as well as subsets: just latencies (the first 8 features) and just intensities (final 9 features). These plots are displayed in Figure 2.3. In each of these plots, points of the same color come from the same user.

The differences between the PCA visualization resulting from the entire set of features and the visualization resulting from the subsets of features are striking. While visible clusters are evident in the subset plots, the clusters are much more distinct in the plot of the combined set of features. Even before we had found a successful classification model these visualizations provided support for the hypothesis that the keystroke intensity data we collected provided information that was both a) useful for distinguishing between users and b) not already present in the latency data.

3 Classification by Keystroke Patterns

We evaluated the performance of several schemes that have been previously tested in keystroke dynamics publications. These models include Euclidian distance, non-weighted probability, and weighted

probability measures¹ as well as Support Vector Machines (SVM) and Multilayer Perceptrons (MLP).² The success of each method seems to vary across publications, depending on the specific nature of each problem and approach. With minimal guidance on where to begin our analysis, we performed a barrage of tests using modified versions of MATLAB built-in tools.

Clustering by K-means produced results with relatively high error rates, steering our approach away from Euclidian distance measures.

We then moved on probabilistic models and explored the use of a Naive Bayes Classifier. We obtained the most satisfying results with a manually adjusted prior favoring the likelihood of seeing an imposter as opposed to the actual user.

Additional tests with SVMs, using a linear kernel, were performed with little to no improvement over the results given by Naive Bayes.

Our most successful scheme used a MLP. Preprocessing and training were done with the aid of the MATLAB *Neural network pattern recognition tool*. Preprocessing steps were limited to scaling the data within the range of $[-1, 1]$. The resulting model included one hidden layer of 10 neurons. It was trained with Scaled Conjugate Gradient Backpropagation using the Mean Squared Error performance function. We split our data into 3 pieces: 70% training, 15% validating, 15% testing.

Most of the techniques we used are standard in the use of MLP's, but we did have to deal with some peculiar convergence issues. Fairly frequently while

¹Authentication via keystroke dynamics. Fabian Monrose and Aviell Rubin. 1997. In Proceedings of the 4th ACM conference on Computer and communications security (CCS '97). ACM, New York, NY, USA, 48-56.

²User authentication through typing biometrics features. L.C.F. Araujo, L.H.R. Sucupira Jr, M.G. Lizarraga, L.L. Ling, J.B.T. Yabu-Uti Sch. of Electr. & Comput. Eng., State Univ. of Campinas, Brazil IEEE Transactions on Signal Processing.

training our network pathological convergence behavior would occur, and our network would either a) fail to converge or b) converge way too soon and perform terribly on the training data. We deal with a) by capping the number of iterations that a network may take while training to 200 iterations. We deal with b) by performing a 'degeneracy' test after the network converges. After convergence we calculate the FAR and FRR error rates using the training and validation data, and if these rates are too high then we retrain the network. Nowhere in this degeneracy test is the testing data used. Eventually, sometimes after a large number of iterations, the network's performance meets our minimal criteria, and only then do we consider the network properly trained. The addition of this degeneracy test radically improves the performance of our classification model.

4 Results

We evaluated our success based upon the minimization of the following error rates, with the specific aim to minimize FAR below 1 %:

General Error Rate: overall frequency of misclassification

False Acceptance Rate (FAR): frequency of classifying a false user as the valid user. High FAR implies that many illegitimate users could access password protected information.

False Rejection Rate (FRR): frequency of the legitimate user being identified as an imposter. High FRR implies that the valid user may need to repeatedly type the password before being verified.

	Latency Only		Latency and Intensity	
	Naive Bayes	Neural Network	Naive Bayes	Neural Network
General	8.6 %	4.8 %	5.7 %	2.3 %
FAR	0.7 %	1.0 %	2.1 %	0.8 %
FRR	94.3 %	60.1 %	47.9 %	22.7 %
		Best Worst		Best Worst
General		2.0% 1.0%		0.5% 6.1%
FAR		0.0% 2.6%		0.0% 2.2%
FRR		5.6% 100%		0.0% 50.0%

Table 4.1: Summary of final error rates

We trained and tested our model 14 times, each time choosing a different user in our dataset to represent the valid user while the other 13 were seen as imposters. Overall error rates were calculated as the average over all 14 users. Best and worst error rates for individual users are shown for our neural network results. Of note is the 100 % failure of a latency-only model to confirm the identity of a user

in the worst case. We also found 100 % worst case FRRs for Naive Bayes trained on latencies only as well as on our full feature set. Also of note is the ability of our best model to identify left out users (those not included in the training set) as imposters with a similar error rate (data not shown).

Ultimately, we found that the MLP described in the previous section performed the best and offered the most flexibility in trading off between FAR and FRR. This flexibility is important in the context of security and authentication problems. A low FAR corresponds to strong security and a low FRR corresponds to convenience. Depending on the context, convenience may be more important in an authentication scheme than extremely stringent security. In other cases the exact opposite could be true. Ideally both of these needs could be met simultaneously, but unfortunately, in any classification scheme there is uncertainty at play, due to noise in the data, changing behavior of the user, or inadequacy in the model, so it may be impossible to maintain both a sufficiently low FAR and FRR. But it is often possible to trade off between the two. The two classifiers that we investigated the most, Naive Bayes and MLP, offer simple ways to make this trade-off. For Naive Bayes we can simply manually adjust the class prior for imposters. For MLP's we can change the threshold at which the network's output is considered to classify the input as coming from the legitimate user.

The neural network whose performance is depicted in the table used a threshold of 0.5, where for each password attempt, the attempt is classified as originating from the legitimate user if the network outputs a value at least 0.5 and from an imposter if the network's output is less than 0.5. This particular threshold offers error rates which are favorable in a general security setting. But, as discussed earlier, different thresholds may be superior in specific situations. The performance of different thresholds are depicted in Figure 4.1.

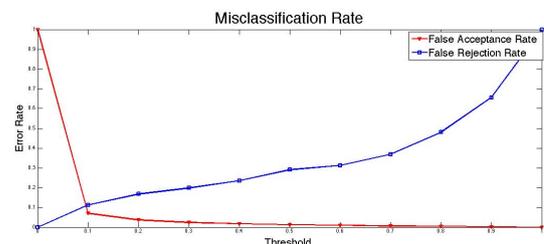


Figure 4.1: FAR and FRR plotted against classification threshold

This plot was generated by varying the MLP's threshold from 0 to 1 and calculating the FAR and FRR at each level. The point at which FAR

and FRR are equal defines the **Equal Error Rate (EER)** which we found to be approximately 10 % at a threshold of 0.1. This data is consistent with previously published work on the use of keystroke latency and intensity (as measured through pressure sensitive keyboards) for user authentication, which has reported EERs between 11 and 40 %.³

5 Conclusion

Our investigation has shown that intensity is highly informative as an additional feature in user authentication through keystroke biometrics. Both the separation between users obtained in PCA analysis as well as error rates in valid user detection were vastly improved by the addition of intensity to the traditionally used latency measures. Previous work has corroborated the efficacy of pressure through different methods. These studies utilized pressure sensitive keyboards and extracted features such as total harmonic distortion, kurtosis, and energy.⁴ Audio signals also carry these characteristics; however, within an audio signal they are often obfuscated by extraneous noise from the user's environment. Peak extraction corresponding to the keystroke intensity presents itself as a simple method to avoid the noise problem and preserve valuable user-specific information.

This paper introduces the use of a laptop's built-in microphone in keystroke intensity estimation, for the purposes of user authentication. Further investigation into this idea should prove the robustness

of the model in several areas. First, data should be collected while the user is typing in environments with varying noise levels. It should be proved that signal isolation or noise reduction methods can be employed to ensure that the extracted peak volumes correspond to the sound of the keyboard. Second, users should be recorded across a number of different sessions to confirm that typing patterns are retained by a single user across time. Third, further exploration of password length and content should be conducted. We have seen that certain character combinations vary more significantly between users and that classification accuracy improves with the length of the password (abbreviated data only from 'andrew' produced much higher error rates than the full password).

To conclude, biometric user authentication is of concern in a variety of areas from password hardening to confirming student ID for the distribution of verified certificates of completion from online course providers.⁵ Our results offer a method of improving the quality and ease of using keystroke dynamics for these purposes.

In other words, intensity is key. :)

6 Acknowledgments

We would like to thank the 14 users who gave their time and intensity for the advancement of this important area of research.

One more thing...

We love you Andrew!

³Pressure-based Typing Biometrics User Authentication Using The Fuzzy ARTMAP Neural Network C. C. Loy, C. P. Lim, and W. K. Lai International Conference on Neural Information Processing, Taiwan, 2005

⁴Keystroke Patterns Classification using the ARTMAP-FD Neural Network C. C. Loy, W. K. Lai, and C. P. Lim International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Taiwan, 2007

⁵<http://edf.stanford.edu/readings/course-announces-details-selling-certificates-and-verifying-identities>