Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 6th, 2011

A Keystroke Biometric System Test-Taker Setup and Data Collection

Vinnie Monaco, Edyta Zych, John Stewart, Charles Tappert,
Tyrone Allman, Mino Lamrabat, Mandar Manohar,
Hassan Poorshatery, Geoffrey Garcia, Elizabeth Teracino, Xiaolu Zhao

Pace University Seidenberg School of CSIS, White Plains, NY 10606, USA

Abstract

Pace University's Seidenberg School of Computer Science and Information Systems (CSIS) has developed the Pace Keystroke Biometric System (PKBS), which can be used for both identifying and authenticating users via their typing rhythms and patterns through the monitoring and capturing of keyboard events. It has the capability to recognize with a high degree of accuracy the typing characteristics that are unique to each individual. Pace's CSIS has been conducting research on this particular form of biometrics for the past eight years. The PKBS consists of three components: the Keystroke Entry System (KES) that collects data over the Internet, the Keystroke Feature Extractor (KFE) that extracts a feature vector from the raw data collected via the KES with the purpose of characterizing the individual's typing dynamics, and the Keystroke Pattern Classifier (KPC) that is used in the authentication process. This research paper focuses on enhancements to the keystroke entry system to support a real-world application to authenticate students taking online tests.

1. Introduction

Keystroke Biometric System (KBS), developed at Pace University, has been enhanced and improved over the last several years. The system was built with the objective of capturing user keystrokes and via a unique k-NN classier, identify and authentic the user [1]. When a user profile is created, that profile can be used to compare typing patterns for a variety of different metrics, giving the computer the ability to actively monitor its user and remove privileges when required. Accurate and effective keystroke biometric technology can help provide a major boost to the security of electronic commerce, and it can help curb identify theft [2]. The goal of this study is to demonstrate how this system can be used in an authentication application to verify the identity of students taking online quizzes or tests that require extensive textual input from the user. This is an important application with student populations in online classes increasing and instructors becoming more concerned about evaluation security and academic integrity [1,6].

KBS is attracting interest because keyboards are readily available and keystroke data-capture is not intrusive and computer users, for work or pleasure, frequently type on a computer keyboard. Second, little capital investment is required with such a system. Third, keystrokes continue to be entered for potential subsequent checking after an authentication phase has verified a user's identity since keystrokes exist as a mere consequence of users using computers. Finally, with more businesses moving to e-commerce, the keystroke biometric in internet applications can provide an effective balance between high security and ease-of-use for customers.

Generally, a number of measurements or features are used to characterize a user's typing pattern. These measurements are typically derived from the raw data of key press times, key release times, and the identity of the keys pressed. From key-press and key-release times a feature vector, often consisting of keystroke duration times and keystroke transition times, can be created [2]. Such measurements can be collected from all users of a system, such as a computer network or web-based system, where keystroke entry is available, and a model that attempts to distinguish an individual user from others can be established.

The keystroke biometric system measures typing characteristics believed to be unique to an individual and difficult to duplicate. It can provide strong authentication to Web-based applications, e-mail and networks [3]. Currently, there is an existing software program called "ThirdFactor" [5], which uses keystroke biometrics to actively authenticate users. In order for this program to operate, it must collect the user's keystroke information for at least one day of

computer usage. After this process, the program creates a user profile which is used for comparing with the current user score. The program continuously computes a score based on the current user's typing patterns [6].

The keystroke biometrics system is still less popular than other forms of biometric authentication because not enough people are familiar with it. The system works especially well for workplaces, where most employees use the same keyboard day after day. The same applies for online courses, in which a student might follow the same course for nine weeks–allowing time for the system to measure his typing patterns–and then take a test that requires strong authentication in the final week.

2. Improved Keystroke Entry System

The KeyStroke Entry System (KES) is used to log and store keystroke data for students taking online tests (Figure 1). Keystoke features captured from the raw data generated by each user are extracted and feature vectors are developed. Reference and test feature vector samples are compared using a k-NN classifier in an attempt to recognize each user.

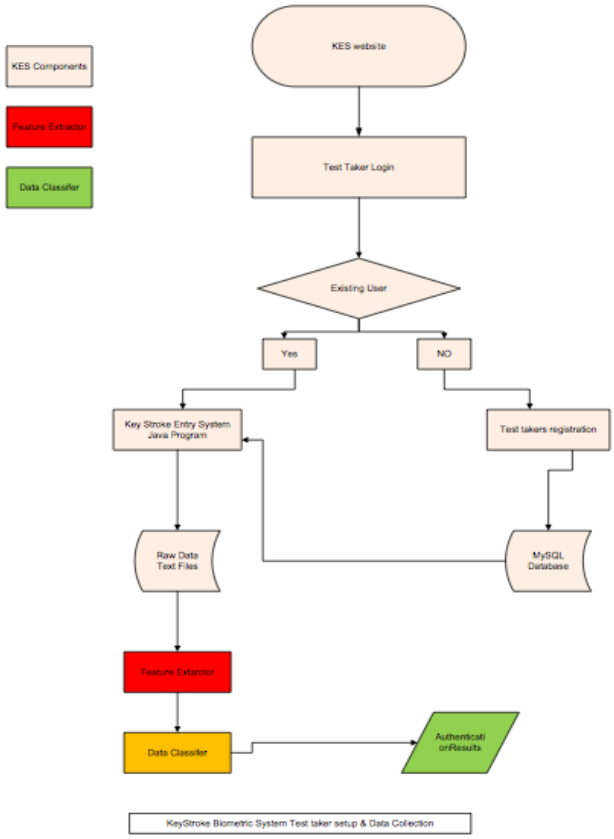


Figure 1. The KES flow chart

The KES is a standalone java application which starts when the test taker navigates to the testing web site and activates the test. Once this is done, authentication is required and the test taker is asked for their login credentials. If a user is already registered in the system, they will login as usual to access a test. If the user has never before registered, they will create a username and password to activate the current test as well as all future tests.

The students login to the Test Takers Interface by using the KES website and register as new users create name and login data, which is stored in a MySQL database and will be retrieved later when the user logs in to take the test. The registered students login to the interface to take the test, and answer the questions, which will be displayed from the database. The KES displays the question while monitoring the text entry screen where all the keystrokes entered by the student are captured without her/his knowledge into separate text files in the database. These text files (the output stage of the KES) are analyzed and compared by using the Biometric Authentication Feature Extractor and Feature (data) Classifier to identify the keystroke patterns of each test taker (students) and authenticate them.

In the new version of the KES significant improvements have been made over prior versions of the system: a new xml data format instead of just plain text; standalone java program instead of an applet; using a test taking format instead of free text or a copying task; and incorporating a user management system for authentication. The changes have been made to improve accuracy and to make the KES a more user-friendly system.

2.1 Data capture

The Keystroke Entry System is a Java based program that can be downloaded and run on the local java enabled machine. The local machine has to be connected to the internet as the application captures data relating to key strokes on a computer and stores a data file of the information on a server. Upon entering the system a registered user can enter his/her first name, last name, username and password to gain access to the system or a new user can register. Once authenticated the user is presented with a test and a series of questions that require at least 200 characters in a plain text entry field.

2.2 Data format

The data collected in this application can be divided into three different categories

A. Keystrokes:
The keystroke data contains information about the single press and release of a key. This includes the entry number, the key pressed, the key code of the key pressed, time pressed, time released, and duration the key was held down. It has this structure:
{entry key keyCode timePressed timeReleased duration}

B. Transitions:
Transition data is a sequence of transitions. A single transition contains the information between two keystrokes. The four transition parameters used are:

- Type 1 = secondKeystroke.timePressed - firstKeystroke.timeReleased
- Type 2 = secondKeystroke.timePressed - firstKeystroke.timePressed
- Type 3 = secondKeystroke.timeReleased - firstKeystroke.timeReleased
- Type 4 = secondKeystroke.timeReleased – firstKeystroke.timePressed

C. Stylometry
This is the actual text that the user types in when answering the questions in the test.

3. Authentication Experiments

In order to determine if a student can be authenticated during an online test, a set of measurements is made on the typing patterns and text input. This results in a vector of features, which may be unique to that student. This vector can be used to train a classifier and build a model to later authenticate new input against.

The online test taking system captures both typing patterns and text input, allowing us to run this information through a feature extractor. The feature vectors obtained from this process can then be used to build a classifier and attempt to authenticate students.

3.1 Feature extractor benchmark results

The feature extractor, as well as the input system, has been completely revised from the original version in order to reflect a more realistic online test scenario and possibly improve authentication performance. In order to test the accuracy of the new feature extractor, benchmark experiments on old data were conducted using 5 data samples (yielding 10 feature difference vectors) from each of 18 training subjects and similar data from 18 different test subjects. Authentication results obtained with the original and new feature extractors (239 features) are summarized in Table 1. The five samples from each of 18 subjects produced 180 within-class and 3825 between-class feature difference vectors (test and train sizes). Because testing was on untrained subjects this is referred to as "weak" training. When testing is performed on trained subjects, training is referred to as "strong."

System	Test Sizes	Train Sizes	FRR	FAR	Performance	kNN
Old Feat Ext	180-3825	180-3825	10.00% (18/180)	3.29% (126/3825)	96.40% (3861/4005)	1
New Feat Ext	180-3825	180-3825	11.67% (21/180)	3.22% (123/3825)	96.40% (3861/4005)	1

Table 1. Old versus new feature extraction results: training on 18 and testing on 18 different subjects.

These results indicate that the revised feature extractor maintains the same accuracy as the original one, with slight differences in FRR and FAR.

3.2 Feature normalization methods

Previously, features were normalized using the minimum and maximum values. Each feature x was normalized to x' using the following formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The revised feature extractor has the option of normalizing each feature using +/- 2 standard deviations as the minimum and maximum values in the previous formula. For each feature x, the following formula is used:

$$x' = \frac{x - \mu_x + 2\sigma_x}{4\sigma_x}$$

with x' clamped in the range 0-1 (<0 set=0, >1 set=1). Table 2 compares these normalization methods on the benchmark data.

Biometric	Test Sizes	Train Sizes	FRR	FAR	Performance	kNN
Keystroke MIN/MAX	180-3825	180-3825	11.67% (21/180)	3.22% (123/3825)	96.40% (3861/4005)	1
Keystroke STD DEV	180-3825	180-3825	7.78% (14/180)	3.87% (148/3825)	95.96% (3843/4005)	1

Table 2. Min/max versus std normalization methods.

The results of the two normalization methods differ only slightly, one showing a better FAR and the other a better FRR.

3.3 Results on new data

A class of 38 students in a spreadsheet modeling

course took four online tests of 10 questions each. Tests took place at approximately two week intervals. Keystroke timing raw data was captured in the new test taking environment and feature extraction using the +/-2 std normalization method was performed as described above. The data were captured in a classroom setting and students were unaware of their keystrokes being logged. Results were obtained from the 38 subjects using 10 samples per subject from the first of the four tests – 20 subjects for training and 18 subjects for testing. Performance results for data collected from the first test are shown in Table 3.

Biometric	Test Sizes	Train Sizes	FRR	FAR	Performance	kNN
Keystroke	810-15300	900-19000	20.25% (164/810)	4.18% (640/15300)	95.01% (15306/16110)	1

Table 3. Performance results on new data.

The false acceptance rate has remained roughly the same as the benchmark results, while the false rejection rate has risen. This led to a slight decrease in performance across all kNN. The increase false rejection rate may be attributed to outlier samples within an individual, as typing patterns between questions on a test may have varied more. This was confirmed when several samples were found to contain no keystrokes at all, or short answers such as "I don't know." In this case, it may not be as relevant whether the individual can be authenticated, since the answer to the test does not contain any useful information for grading. These scenarios may have to be handles separately in the next revision to the feature extractor, to ensure the quality of the keystroke data being processes.

3.4 Results on longer data samples

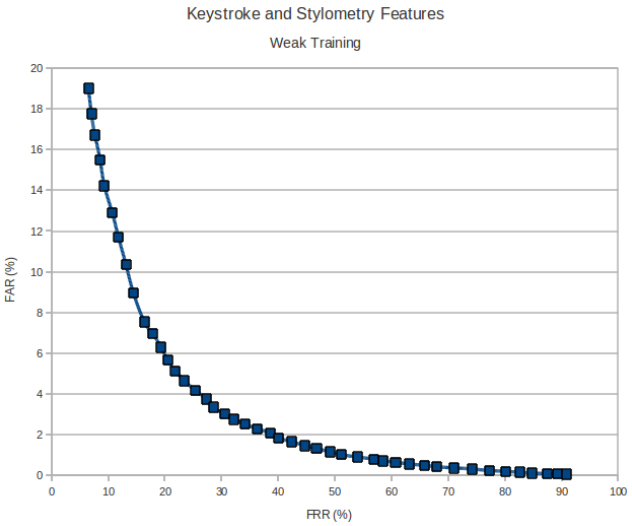
In an online test-taking scenario, it is not possible to ensure that all data being captured is accurate and of high quality (compared to previous experiments in which users were aware of their keystrokes being captures and could be prompted to provide lengthy responses [3]). Short answers such as "I don't know" in the previous experiment are clearly insufficient for reasonable authentication. Therefore, pairs of samples were concatenated to provide longer data samples. Each sample is then much more likely to contain significant keystroke and stylometry information.

Samples were collected on two separate dates from the same group of students. The table below summarizes both strong and weak on 760 samples collected from 38 subjects, which each contain an average of 100 to 200 words in response to a question asked in an online test:

Training	Biometric	Test Sizes	Train Sizes	FRR	FAR	Performance	kNN
Strong	Keystroke	1665-66600	1755-74100	18.80% (313/1665)	8.35% (5559/66600)	91.40% (62393/68265)	1
Weak	Keystroke	2810-69200	3010-69000	38.93% (1094/2810)	4.66% (3227/69200)	94.00% (67689/72010)	1

Table 4. Strong vs Weak Performance Results.

The weighted ROC curve for k=10 with weak training, as described in [3], is shown below:



Graph 1. ROC curve for k=10

It can be seen that an equal error rate is achieved at about 12% FAR and FRR. This is reasonable compared to the baseline experiments, considering the larger population and errors due to incomplete answers given on a test.

For each student, two samples were combined to produce one sample twice as long. The total number of samples was reduced 380 (190 test and 190 train), with each sample containing an average of 200 to 400 words. Features were then taken over this reduced set of data. This is summarized in the table below:

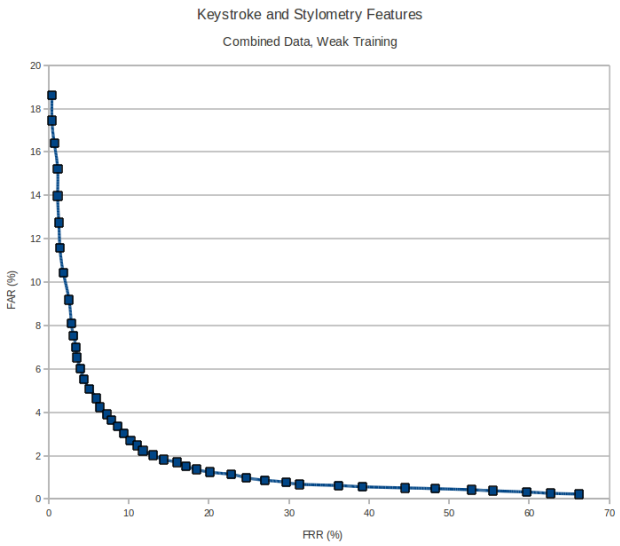
Training	Biometric	Test Sizes	Train Sizes	FRR	FAR	Performance	kNN
Strong	Keystroke & Stylometry	380-17575	380-17575	3.42% (13/380)	9.05% (1591/17575)	91.07% (16351/17955)	1
Weak	Keystroke & Stylometry	705-17250	755-17200	15.04% (106/705)	4.78% (825/17250)	94.81% (17024/17955)	1

Table 5. Strong vs Weak combined samples.

In this case, weak training has outperformed the strong training. The combined samples have reduced the FRR significantly, while only slightly increasing the FAR. This is due to a wider range of data for each sample. A student may have different typing

patterns depending on the question being asked on a test. Combining these two samples results in a more accurate feature vector for that user, but may also be similar to other users, which explains the higher FAR. The unweighted ROC curve for weak training in this experiment is shown below:

Graph 2. ROC curve for k=10



The equal error rate occurs at about 5%, indicating an improvement in the quality of the data. Combining samples has significantly decreased the chances of processing incomplete samples without the loss of any information that would have occurred by omitting samples entirely.

Conclusion

Keystroke biometric is an inexpensive, yet effective method of user identification and authentication. The PKBS, if developed further, could be particularly ideal for student online testing when embedded within a browser and customized per the institution utilizing the system.

While the main concern is currently surrounding academic integrity during online testing, this sort of authentication could be used for the same reasons when training and orientation examinations are administered in a business setting. Or consequently, to monitor email transactions over a company server as a preventative measure for potential email misuse and/or scandal.

Future Work

The ability to turn individual features on and off will help us better select the features to be used for authentication. This process may be automated by selecting features with a higher frequency, which carry more information about the user's typing patters. Efficiency has also become a concern as more data is collected. The algorithm used to classify and authenticate samples is exponential in time, taking up to 24 hours to complete on a desktop machine processing 800 samples. In order build a system that can handle large amounts of data, the algorithm will have to be made multi- threaded in order to take advantage of a distributed environment.

References

[1] Ed Dante, The Shadow Scholar, The Chronicle Review, The Chronicle of Higher Education, November 12, 2010.

[2] R.Joyce and G.Gupta, “Identity Authentication Based on Keystroke Latencies,” Communications of the ACM Volume 33, Issue 2, 1990, pp.168-176.

[3] A.Monrose and A.D. Rubin, “Keystroke Dynamics as a Metric for Authentication,” Future Generation Computer Systems, Volume 16, Issue 4, 2000, pp.331-359.

[4] A.Peacock, X. Ke, and M. Wilkerson “Typing Patterns: A Key to User Identification” IEEE Security and Privacy, volume 2, issue 5, pp.40-47, September-October 2004.

[5] http://www.thirdfactor.com/2011/03/16/ keystroke-dynamics-secure-computer-access

[6] M. Scheible and L. McNabb, “IAM Online: Hot Topics and Current Issues in ... HEOA on Distance Education,” Web Conference, March 2010,

[7] S. Yoon, S.-S. Choi, S. Cha, Y. Lee, and C. C. Tappert., "On the individuality of the iris biometric," *Int. J. Graphics, Vision & Image Proc.*, vol. 5, pp. 63-70, 2005.