

An Enhanced Keystroke Biometric System and Associated Studies

Tarjani Buch, Andreea Cotoranu, Eric Jeskey, Florin Tihon, Mary Villani

Seidenberg School of CSIS, Pace University, New York, U.S.A.

{tb43439w, ac73135w, ej01113w, ft26714w}@pace.edu, villanmv@famingdale.edu

Abstract

The Keystroke Biometric System at Pace University identifies subjects based on long-text (about 650 keystrokes) samples. The system consistently identifies subjects using the same keyboard type (desktop or laptop) and entry mode (copy task or free text input) with degrees of accuracy ranging from 98% to 100%. The current work enhances the previously developed keystroke biometric identification system and performs additional studies. Additional identification experiments support previously documented accuracy findings, and new authentication studies indicate promise for authentication application of the system. The feature extractor component is modified to facilitate these and future experiments. Additional data samples are captured for controlled longitudinal studies to verify that reasonable accuracy can be maintained with durations of several weeks between data captures.

1. Introduction

Keystroke biometric systems measure characteristics believed to be unique to an individual and difficult to duplicate [3]. The Keystroke Biometric System developed at Pace University is a faculty originated project. The system has been enhanced by graduate students with the purpose of supporting faculty and doctoral student research in the area of pattern recognition. The system was developed for long-text input applications such as identifying perpetrators of inappropriate email or imposter online test takers [3].

2. Historical Review

The Keystroke Biometric System project started in 2004 and has since gone through four project iterations with different graduate student teams. The system

consists of a Java applet which collects raw keystroke data over the Internet. Following data collection, long-text-input features are extracted, and a pattern classifier makes identification decisions.

The first project was a user identification feasibility study that developed the Java applet and initiated the work in this area, primarily using the statistical SAS software for the development of the feature extraction and classification programs. The second project generated a complete system in Java, which was not user friendly and not successfully applied. An improved third system, also in Java, was used in a feasibility study of a copy typing task. It was then enhanced to support a study involving 118 subjects, and using two input modes (copy and free-text input) and two keyboard types (desktop and laptop keyboards). The fourth system emphasized fallback models which deal with missing or insufficient statistical information, such as too few instances of infrequent letter keys. Two fallback models were developed: one based on linguistic and the other on touch-typing principles. The former substitutes incomplete or insufficient data with more generalized grammatical data while the latter substitutes incomplete or insufficient data with generalized data based on the geography of a standard computer keyboard. These project iterations and related research resulted in several external publications as well as in two doctoral dissertations (see [3] and references therein). The current fifth project extends the earlier work as described below.

3. Keystroke System and Enhancements

The transition from the fourth to the fifth system iteration focuses on the enhancement of the feature extraction component so that feature data can be used not only by the pattern classifier of this system but also by other classifiers.

The keystroke biometric system consists of three main components: raw keystroke data collection,

feature extraction, and pattern classification. The system employs a Java applet to collect keystroke data over the Internet, long-text input features are then extracted, and a pattern classifier makes identification decisions. Figure 1 presents the process and data flow for the keystroke biometric system.

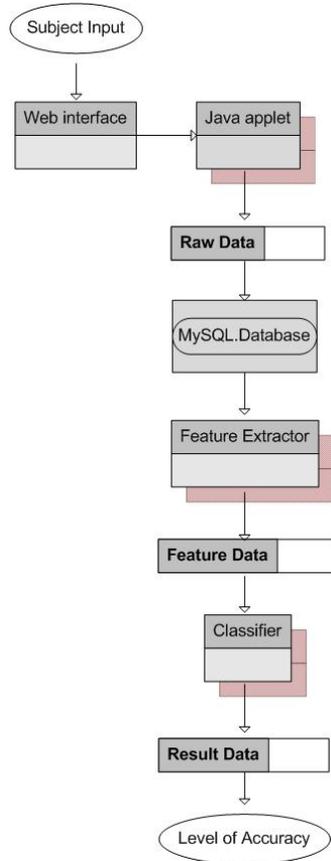


Figure 1: Keystroke biometric system process and data flow diagram.

3.1. Raw Keystroke Data Collection

A subject participating in the data collection process is required to register at his/her first visit to the system web site. The registration process captures demographic data including: subject's first and last name, email, country of schooling, nationality, native language, age, level of education, the mostly used keyboard type, handedness, computer manufacturer, how subjects learned to type, and typing approach. To insure that all subjects have submitted demographic data, each time a subject returns to the web site, he/she is asked for his/her first and last name. A query is then run looking for the subject in the database. If first name and last name are found, the subject is taken

directly to the activity selection Web form and is asked about the keyboard style (desktop or laptop) he/she is using and about what entry task he/she would like to perform – a data copy task or an extemporaneous task. However, the fifth system iteration now asks all subjects to complete data entry samples for each of the following tasks: copy-laptop, copy-desktop, free-text laptop, free-text desktop. This creates a more complete basis for comparison experiments.

A Java applet enables the collection of keystroke data over the Internet (Appendix 1). A submission number is automatically initialized and then incremented after each sample submission. The subject can then start typing his/her data sample. If the subject is interrupted during data entry, the “Clear” button will blank all fields, except name and submission number, and allow the user to redo the current entry. The application raw data file records the following information for each entry: key's character, key's code text equivalent, key's location (1 = standard, only one key location; 2 = left side of keyboard; 3 = right side of keyboard), time the key was pressed (milliseconds), time the key was released (milliseconds), and the number of left-mouse clicks, right-mouse clicks, and double-left-mouse click events during the session. Upon pressing submit, a raw data text file delimited by the “~” character is generated [3].

Previous studies by Villani et al. [3] indicated that, although all subjects were invited to participate in all four entry tasks (copy-laptop, copy-desktop, free-text laptop, free-text desktop), time or equipment limitations led some to opt for only two while others participated in three or four entry tasks. Ultimately only 36 subjects completed all four entry tasks. Furthermore, in these studies the timing of the input samples was not controlled and over half of these 36 subjects entered their data samples in one sitting.

Therefore, for the current and subsequent experiments all subjects are asked to complete five data entries for each of the four entry tasks and submit them within a two day interval. In order to address the need for controlled timing in data collections, those subjects who would like to complete additional sets of four entry tasks are asked to leave approximately two weeks between sets.

For the current studies additional data samples were collected to support a controlled longitudinal study to determine whether reasonable accuracies can be maintained with durations of several weeks between data captures. Data collection was scheduled as per Table 1. The time intervals T_0 , T_1 , and T_2 , indicate two week intervals.

Entry Task	Time		
	T ₀	T ₁	T ₂
Copy-Laptop	5	5	5
Copy-Desktop	5	5	5
Free-Text Laptop	5	5	5
Free-Text Desktop	5	5	5
Total # of Samples	20	20	20

Table 1: Data collection schedule - one subject.

3.2. Feature Extraction

The feature extraction component, a Java application, reads all raw keystroke data files from a local directory. One string of data is created from raw data files and stored in a vector. The vector is read in ascending order from index zero to index N, where N is the number of raw keystroke data files. A second vector is instantiated to track the frequency of each feature detected from the raw data. The lower level features are simply the keys pressed. The higher level features are dependant on the fallback method used in the analysis: linguistic or touch type. The features characterize the typist's key-press, duration times, transition times in going from one key to the next, the percentages of usage of the non-letter keys and, or mouse clicks, and the typing speed. There are a total of 254 features (see [3] for details).

In order to support additional identification and new authentication experiments, the feature extractor was modified to output data in a standard format with fields in comma delimited records as follows (Figure 2):

- the first record contains the name and description of the file;
- the second record contains the number of samples or pattern instances;
- the remaining number of pattern instances are contained in a record with the following fields: ID data (e.g. name/gender/age), person's application-related information (e.g. handedness), equipment related information (keyboard type), task performed (copy text or free text), number of attributes/measurements, and sequence of feature values normalized into the range 0-1;

Name and File Description	Keystroke Biometric Data - Desktop Copy - Created 11/26/2007							
Number of Pattern Instances	180							
FirstName	LastName	Gender	Age	Handedness	Keyboard Type	Task Type	Number of Features	Sequence of Features
JANE	DOE	Female	20 - 29	Right	DESKTOP	FABLE 1	254	0.15164 0.15164

Figure 2: Feature vector data output.

3.3. Pattern Classification

After all features have been extracted into a "features" file, the data is ready to be classified in an attempt to identify the author. The identification is a measure of the sum of the Euclidean distances of all the collected features.

The analysis can be done using one of two methods. With the first method, "train-on-one" or "leave-one-out," one features file is used and classification occurs by pulling out each data entry and comparing it to all the other data in the features file. Classification is successful if the Euclidean distance is least with respect to another data entry by the same author. With the second method of classification, "train on one and test on the other," the classifier is trained on one features file and it attempts to match the data from a second features file to those in the training file. Again, a match in this case is successful when, given the same author, the Euclidean distance is least between the data being tested and the data in the training file.

4. Experiments

Biometric systems have identification and authentication applications. In identification applications a user is identified from within a population of n users (one of n responses). In authentication (verification) applications, a user is either accepted or rejected (binary response, *yes* you are the person you claim to be or *no* you are not).

Previous projects approached identification related problems reasoning that high recognition accuracy would yield system success. The current project supports both identification and authentication experiments.

For classification purposes, the system must be trained before it becomes usable, so data was separated into two sets: one set for training the system to create boundaries in feature space and one for testing the system to determine its accuracy.

One way to support identification and authentication experiments is by using data mining techniques. For this purpose, the project was split between front-end and back-end components. The front-end component managed data gathering and feature extraction while the back-end component managed classification processing.

4.1. Identification Experiments

Experiments were conducted by Villani et al. [3] on 36 subjects who completed all four data entry tasks: desktop-copy, desktop-free, laptop-copy, laptop-free. Experimental results verified that the system identifies subjects with a high degree of accuracy if the subjects use the same keyboard type (desktop or laptop) and entry mode (copy task or free text task). These experiments tested optimal conditions (sub-experiments a and b), combined conditions (sub-experiment c) and less optimal conditions (sub-experiments d and e). All the experiments used the linguistic fallback model. Experiments for optimal and combined conditions used the “leave-one-out” classification method. These experiments revealed a very high level of accuracy, ranging between 98.3% and 100%. Experiments for the less optimal conditions used the “train on one and test on the other” classification method. These experiments revealed a level of accuracy ranging between 50.3% and 91.7%. A complete set of experimental results on these subjects is presented in Table 2 [3].

Experiment	Train	Test	Accuracy
1. Copy Task (36 subjects)	a Desktop	Desktop	99.4%
	b Laptop	Laptop	100.0%
	c Combined	Combined	99.5%
	d Desktop	Laptop	60.8%
	e Laptop	Desktop	60.6%
2. Free Text (36 subjects)	a Desktop	Desktop	98.3%
	b Laptop	Laptop	99.5%
	c Combined	Combined	98.1%
	d Desktop	Laptop	59.0%
	e Laptop	Desktop	61.0%
3. Desktop (36 subjects)	a Copy	Copy	99.4%
	b Free Text	Free Text	98.3%
	c Combined	Combined	99.2%
	d Copy	Free Text	89.3%
	e Free Text	Copy	91.7%
4. Laptop (36 subjects)	a Copy	Copy	100.0%
	b Free Text	Free Text	99.5%
	c Combined	Combined	98.9%
	d Copy	Free Text	86.2%
	e Free Text	Copy	91.0%
5. Different Mode/Keyboard (36 subjects)	a Lap Free	Lap Free	99.5%
	b Desk Copy	Desk Copy	99.4%
	c Combined	Combined	98.6%
	d Desk Copy	Lap Free	51.6%
	e Lap Free	Desk Copy	58.0%
6. Different Keyboard/Mode (36 subjects)	a Desk Free	Desk Free	98.3%
	b Lap Copy	Lap Copy	100.0%
	c Combined	Combined	98.9%
	d Lap Copy	Desk Free	50.3%
	e Desk Free	Lap Copy	52.1%

Table 2: Summary of results for 36-subject identification experiments (from [3]).

For this study, these experiments were reproduced for a four subject set of data collected according to the

enhanced process described in Table 1. We only reproduced those experiments that test the less optimal conditions (sub-experiments d and e). For the new experiments, subjects completed all four entry tasks, and each subject submitted five samples per task at three distinct time intervals, T_0 , T_1 and T_2 . A total of 12 experiments were run through the “train on one and test on the other” classification method, using the linguistic fallback model, as per the experiment summary described in Figure 3.

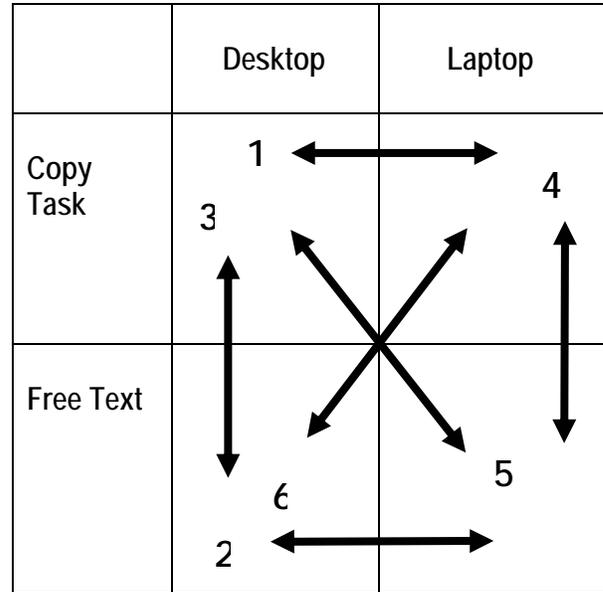


Figure 3: Summary of experimental design [3].

The first hypothesis was that results from running these experiments on the data sets from the four new subjects will reveal similar degrees of accuracy to the experiments run on the 36 subjects. The degrees of accuracy might be even higher in the four-subject experiments, given that the subject population is very small. The second hypothesis was that high degrees of accuracy will be maintained over time from T_0 , through T_1 , and T_2 .

The results of our experiments supported our first hypothesis and revealed, as expected, much higher degrees of accuracy as compared to the 36-subject identification experiments. These results are most likely attributed to the very small sample size (4 subjects).

Results also confirmed the second hypothesis by revealing that high degrees of accuracy can be maintained over time. The degree of accuracy did not decrease significantly over time, and in some of the experiments it did not decrease at all. Out of twelve experiments, five maintained 100% accuracy from T_0 to T_1 and further from T_0 to T_2 . In five of the

experiments, the level of accuracy decreased from T_0 to T_1 to increase again at T_2 . In one experiment, the level of accuracy increased from T_0 to T_1 and then decreased again at T_2 . Furthermore, in one experiment, the level of accuracy increased from T_0 to T_1 and further from T_0 to T_2 . A complete set of experimental results on these subjects is presented in Table 3.

Experiment		Train/Test	Accuracy		
			$T_0 - T_0$	$T_0 - T_1$	$T_0 - T_2$
1. Copy Task (4 subjects)	d	Desktop/Laptop	100%	85%	100%
	e	Laptop/Desktop	100%	95%	100%
2. Free Text (4 subjects)	d	Desktop/Laptop	100%	100%	100%
	e	Laptop/Desktop	100%	100%	100%
3. Desktop (4 subjects)	d	Copy/Free Text	85%	95%	85%
	e	Free Text/Copy	100%	100%	100%
4. Laptop (4 subjects)	d	Copy/Free Text	100%	100%	100%
	e	Free Text/Copy	100%	90%	100%
5. Different Mode/Keyboard (4 subjects)	d	Desk Copy/Lap Free	90%	75%	100%
	e	Lap Free/Desk Copy	80%	95%	100%
6. Different Keyboard/Mode (4 subjects)	d	Lap Copy/Desk Free	95%	100%	95%
	e	Desk Free/Lap Copy	100%	100%	100%
Average			96%	95%	98%

Table 3: Summary of results for 4-subject identification experiments.

The 36-subject data set has also been used for identification experiments using data mining techniques. The experiment used Weka, a data mining tool, with the k-nearest-neighbor (IBk) algorithm with $k=1$ and the “leave-one-out” procedure. The results are comparable to the ones presented in Table 2 [2]. One exception was the experiment which involved training on a laptop with a copy task and testing on a laptop with a free text task; this experiment revealed a very low 18.9% accuracy rate as compared to the 86.2% in Table 2.

Experiment	Train	Test	Accuracy
Copy Task (36 subjects)	Desktop	Laptop	83.34%
	Laptop	Desktop	51.67%

Free Text (36 subjects)	Desktop	Laptop	40.56%
	Laptop	Desktop	52.84%
Desktop (36 subjects)	Copy	Free Text	47.72%
	Free Text	Copy	51.11%
Laptop (36 subjects)	Copy	Free Text	18.89%
	Free Text	Copy	57.78%
Different Mode/Keyboard (36 subjects)	Desk Copy	Lap Free	31.67%
	Lap Free	Desk Copy	55.56%
Different Keyboard/Mode (36 subjects)	Lap Copy	Desk Free	38.07%
	Desk Free	Lap Copy	54.45%

Table 4: Summary of results for 36-subject identification experiments (from [2]).

The identification experiments on the 4-subject data set focused on determining degrees of accuracy from a longitudinal perspective. These experiments were ran through Weka, using the IBk algorithm with $k=1$ and the “leave-one-out” procedure. In these experiments T_0 data set was used for training while the T_1 and T_2 data sets were used for testing [2].

Train	Test	Type	Accuracy with Data Mining (Weka)	Accuracy with Keystroke Biometric System
T_0 (5 samples from each of 4 subjects)	T_1 (5 samples from each of 4 subjects)	Copy Desk	95%	100%
		Free Desk	100%	100%
		Copy Lap	100%	100%
		Free Lap	85%	100%
T_0 (5 samples from each of 4 subjects)	T_2 (5 samples from each of 4 subjects)	Copy Desk	80%	90%
		Free Desk	100%	100%
		Copy Lap	100%	100%
		Free Lap	100%	100%

Table 5: Summary of results for 4-subject identification experiments [2].

4.2. Authentication Experiments

The same two sets of data have also been used for authentication experiments. Experiments on the 36-subject data set were run through Weka, using the k-nearest-neighbor (IBk) algorithm with $k=1$ on the

dichotomy data [1], and the “leave-one-out” procedure [2]. The degrees of accuracy ranged from 62.3%, when training on a laptop with copy text and testing on a laptop with free text, to 97.3% when training on a laptop with free text and testing on a laptop with copy text. To be noticed is that these degrees of accuracy, the highest and the lowest, were both achieved for those experiments involving a laptop keyboard. A complete set of experimental results on the 36-subject data set is presented in Table 6 [2].

Experiment	Train	Test	Accuracy
Train (18 subjects) Test (18 subjects)	Desk Copy	Desk Copy	87.94%
	Desk Free	Desk Free	90.24%
	Lap Copy	Lap Copy	91.03%
	Lap Free	Lap Free	92.06%
Copy Task (36 subjects)	Desktop	Laptop	94.77%
	Laptop	Desktop	80.81%
Free Text (36 subjects)	Desktop	Laptop	62.56%
	Laptop	Desktop	93.10%
Desktop (36 subjects)	Copy	Free Text	82.40%
	Free Text	Copy	82.44%
Laptop (36 subjects)	Copy	Free Text	62.33%
	Free Text	Copy	97.33%
Different Mode/Keyboard (36 subjects)	Desk Copy	Lap Free	62.44%
	Lap Free	Desk Copy	93.37%
Different Keyboard/Mode (36 subjects)	Lap Copy	Desk Free	76.81%
	Desk Free	Lap Copy	89.77%

Table 6: Summary of results for 36-subject authentication experiments (from [2]).

The authentication experiments on the 4-subject data set focused on determining degrees of accuracy from a longitudinal perspective. These experiments were run through Weka, using the IBk algorithm with $k=1$ on the dichotomy data [1]. In these experiments, the T_0 data set was used for training while the T_1 and T_2 data sets were used for testing [2]. The results yielded once again high degrees of accuracy ranging from 88.9% to 100%. These experiments show that high degrees of accuracy can be maintained over time. A complete set of experimental results on the 4-subject data set is presented in Table 7.

Train	Test	Type	Accuracy
T_0 (5 samples from each of 4 subjects)	T_1 (5 samples from each of 4 subjects)	Copy Desk	95.79%
		Free Desk	96.32%
		Copy Lap	91.58%
		Free Lap	92.11%
T_0 (5 samples from each of 4 subjects)	T_2 (5 samples from each of 4 subjects)	Copy Desk	88.95%
		Free Desk	98.42%
		Copy Lap	100.00%
		Free Lap	93.68%

Table 7: Summary of results for 4-subject authentication experiments (from [2]).

5. Conclusions and Recommendations

Previous experimental results have indicated high degrees of accuracy in identifying subjects based on long-text input, especially under the conditions when the same keyboard type is used or the same data entry task is performed. In new experiments, our keystroke biometric system outputs feature vector data in a format that enhances identification processing even further and shows promising for authentication processing.

Although results of current studies support previous experimental results, it is recommended that more raw data be collected following the previously discussed data collection schedule involving two week intervals between data captures. Running experiments with a larger data pool collected under the above conditions should provide stronger evidence relative to the success of the keystroke biometric system for identifying and eventually for authenticating subjects. It would also provide more insight into how accuracy evolves from one data collection session to another over time.

Nevertheless, all experimental results are promising in that the system has the capability of solving identification problems and the potential for solving authentication problems.

7. References

- [1] S. Bharati, R. Hassem, R. Khan, M. Ritzmann and A. Wong, "Biometric Authentication System Using the Dichotomy Model", *Proc. CSIS Research Day, Pace Univ., May 2008*.
- [2] C. Eusebi, C. Gliga, D. John, and A. Maisonave, "A Data Mining Study of Mouse Movement, Stylometry, and Keystroke Biometric Data", *Proc. CSIS Research Day, Pace Univ., May 2008*.
- [3] M. Villani, C. Tappert, G. Ngo, J. Simone, H. St. Fort, and S. Cha, "Keystroke Biometric Recognition Studies on Long-Text Input under Ideal and Application-Oriented Conditions", *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop, February 2006*.

Appendix 1 : Java Applet

```
import javax.swing.*;
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
//import java.security.*;
//import java.rmi.*;

public class KeySpeed16_Applet extends
Applet
implements ActionListener
{
    private int APPLETT_WIDTH = 700;
    private int APPLETT_HEIGHT = 420;

    private KeySpeed16 keySpeed;

    public void init()
    {
        java.net.URL url =
this.getDocumentBase();
        String query = url.getQuery();

        //query = "Hug&Hort&Fable&l&kb&pc";
//
test purposes

        try {
            String[] user =
query.split("&");
            keySpeed = new
KeySpeed16(user[0].toUpperCase(),
user[1].toUpperCase(),

            user[2].toUpperCase(),
user[3].toUpperCase(),

            user[4].toUpperCase());
            add (keySpeed);
            setSize (APPLETT_WIDTH,
APPLETT_HEIGHT);
        }
        catch (Exception e) {

        }
    }

    public void actionPerformed
(ActionEvent event)
    {
        //System.setSecurityManager(new
RMISecurityManager());
    }
}
```