Proceedings of Student/Faculty Research Day, CSIS, Pace University, May 5th, 2006

Keystroke Biometric Recognition Studies on Long-Text Input under Ideal and Application-Oriented Conditions

Mary Villani, Charles Tappert, Giang Ngo, Justin Simone, Huguens St. Fort, and Sung-Hyuk Cha
School of CSIS, Pace University, Pleasantville, New York, 10570

Abstract

A long-text-input keystroke biometric system was developed for applications such as identifying perpetrators of inappropriate e-mail or fraudulent Internet activity. A Java applet collected raw keystroke data over the Internet, appropriate long-text-input features were extracted, and a pattern classifier made identification decisions. Experiments were conducted on a total of 118 subjects using two input modes – copy and free-text input – and two keyboard types – desktop and laptop keyboards. Results indicate that the keystroke biometric can accurately identify an individual who sends inappropriate email (free text) if sufficient enrollment samples are available and if the same type of keyboard is used to produce the enrollment and questioned samples. For laptop keyboards we obtained 99.5% accuracy on 36 users, which decreased to 97.9% on a larger population of 47 users. For desktop keyboards we obtained 98.3% accuracy on 36 users, which decreased to 93.3% on a larger population of 93 users. Accuracy decreases significantly when subjects used different keyboard types or different input modes for enrollment and testing.

1. Introduction

The goal of this study is to prove or disprove the distinctiveness of an individual's keystroke pattern, on long passages, when conditions are not ideal (different entry task, different keyboards).

This paper concerns an identification (one-of-n) application of the keystroke biometric for long-text input. A potential scenario for this application is a company environment in which there has been a problem with the circulation of inappropriate (unprofessional, offensive, or obscene) e-mail from easily accessible desktops in a work environment, and it is desirable to identify the perpetrator. This system could also be used in an authentication application to verify the identity of students taking online quizzes or tests, an important application with the student population of online classes increasing and instructors becoming more concerned about evaluation security and academic integrity.

The keystroke biometric is appealing for several reasons. First, it is not intrusive and computer users, for work or pleasure, frequently type on a computer keyboard. Second, it is inexpensive since the only hardware required is a computer. Third, keystrokes continue to be entered for potential subsequent checking after an authentication phase has verified a user's identity (or possibly been fooled) since keystrokes exist as a mere consequence of users using computers [8]. 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 [19].

Keystroke biometric systems measure typing characteristics believed to be unique to an individual and difficult to duplicate [2, 9, 10]. There is a commercial product, BioPassword currently used for hardening passwords in existing computer security schemes [15], however this is on short entry. Keystroke Biometrics is one of the less-studied biometrics and researchers tend to collect their own data, so few studies have compared recognition techniques on a common database. Nevertheless, the published literature is optimistic about the potential of keystroke dynamics to benefit computer system security and usability [18].

Previous work follows the most commonly adopted metrics to evaluate a biometric system's authentication accuracy are the False Reject Rate (FRR) and the False Accept Rate (FAR) that respectively correspond to the two popular metrics of sensitivity and specificity [2, 7, 12]. Early work of Legett and Williams [13] showed that keystroke digraph latencies had potential for a static identity verifier at login time, as well as a dynamic identity verifier throughout a computer session, and Leggett, et al. [13] conducted similar experiments, reporting 5.0% FAR and 5.5% FRR on a long string of 537 characters. D'Souza's experiment weighted the latencies to reduce false acceptances [4]. Brown and Rogers [3] and Obaidat and Sadoun [15] used short name strings for user verification. Dynamic shuffling was also evaluated as a process applied to training samples for neural networks as a means of enhancing sample classification and reducing false acceptance and rejection rates during keystroke analysis [3]. Recent work by Gunnetti and Picardi [8] suggest that if short inputs do not provide sufficient timing

information, and if long predefined texts entered repeatedly are unacceptable, we are left with only one possible solution, which is using the typing rhythms users show during their normal interaction with a computer; in other words, deal with the keystroke dynamics of *free text.*

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 [19]. 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. For short input such as passwords, however, the lack of sufficient measurements presents a problem because keystrokes, unlike other biometric features, convey a small amount of information. Moreover, this information tends to vary for different keyboards, different environmental conditions, and different entered texts [8]. For these reasons we focus our studies on long text input where more information is available.

This paper extends previous work on a long-text keystroke biometric system that showed the effectiveness of the system under ideal conditions in which the users input prescribed texts, used the same type of keyboard for enrollment and testing, and knew that their keystroke data were being used for identification purposes [1]. In this paper, we implement an improved system (more features and appropriate handling of statistical computations for small sample sizes) and obtain experimental results on more subjects under ideal conditions and extend these results to less favorable conditions where the users enter arbitrary texts, use different types of keyboards for enrollment and testing.

The remainder of the paper is organized as follows. Section 2 describes our keystroke biometric system, having components for data capture, feature extraction, and classification. Section 3 describes the experimental design and section 4 presents the experimental results and conclusions.

2. Keystroke Biometric System

The Keystroke Biometric System consists of three components: raw keystroke data collection, feature extraction, and pattern classification.

2.1. Data Capture

A Java applet was developed to enable the collection of keystroke data over the Internet (Figure 1). The user is required to type in his/her name, although no data is captured on this entry. Also, the submission number is automatically incremented after each sample submission, so the subject can immediately start typing the sample to be collected. If the user 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.

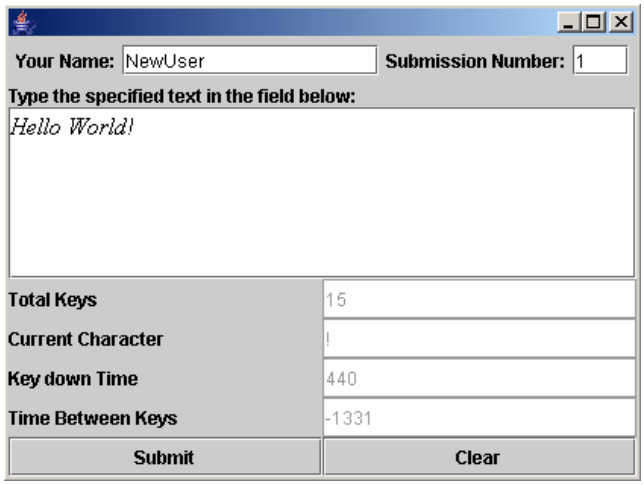

Figure 1: Java applet for data collection.

The raw data file recorded by the application contains the following information for each entry:

- key's character
- key's code text equivalent
- key's location (1 = standard, 2 = left, 3 = right)
- time the key was pressed (milliseconds)
- time the key was released (milliseconds)
- number of left-mouse-click, right-mouse-click, and double left-mouse-click events during the session (note that these are events in contrast to key presses)

Upon pressing submit, a raw-data text file is generated, which is delimited by the '~' character. The aligned version of the raw data file for the "Hello World!" example is shown in Figure 2.

NewUser	Submission 1				
Entry #	Key	Keycode	Location	Press	Release
Num 1	?	Shift	2	1114450735680	1114450736962
Num 2	H	H	1	1114450735991	1114450736311
Num 3	e	E	1	1114450737653	1114450738144
Num 4	l	L	1	1114450738735	1114450739256
Num 5	l	L	1	1114450739786	1114450740277
Num 6	o	O	1	1114450740998	1114450741399
Num 7		Space	1	1114450742090	1114450742420
Num 8	?	Shift	2	1114450743542	1114450745004
Num 9	W	W	1	1114450743872	1114450744263
Num 10	o	O	1	1114450745755	1114450746216
Num 11	r	R	1	1114450747017	1114450747437
Num 12	l	L	1	1114450748138	1114450748549
Num 13	d	D	1	1114450749310	1114450749771
Num 14	?	Shift	2	1114450751373	1114450753776
Num 15	!	1	1	1114450752445	1114450752885
Left Clicks	0				
Right Clicks	0				
Double Clicks	0				

Figure 2: Aligned raw data file for "Hello World!"

2.2. Feature Extraction

The system extracts a feature vector from the information in a raw data file. The features are statistical in nature and specifically designed to characterize an individual's keystroke dynamics over writing samples of 200 or more characters. Most of these features are averages and standard deviations of key press duration times and of transition times between keystroke pairs, such as digraphs [15, 17]. We measure the transitions between keystrokes two ways: from the release of the first key to the press of the second, t_1, and from the press of the first to the press of the second, t_2 (Figure 3). While the second measure, t_2, is always positive because this sequence determines the keyboard output, the first measure, t_1, can be negative (see Figure 3). We refer to these two measures of transition time as type-1 and type-2 transition features.

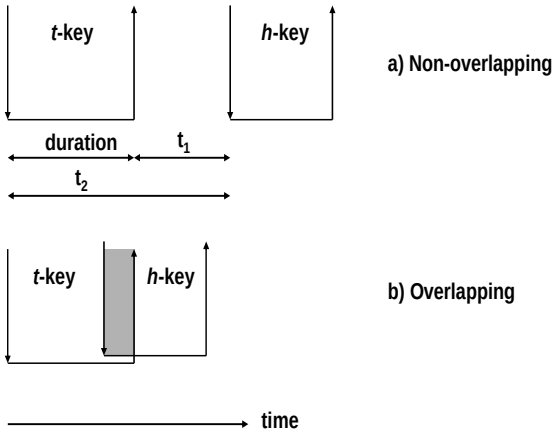

Figure 3: A two-key sequence (*th*) showing the two transition measures: t_1 = *press* time of second key – *release* time of first, and t_2 = *press* time of second key – *press* time of first. A keystroke is depicted as a bucket with the down arrow marking the press and the up arrow the release time. Part a) non-overlapping keystroke events (t_1 positive), and b) overlapping keystroke events where the first key is released after the second is pressed (t_1 negative).

While key press duration and transition times are typically used as features in keystroke biometric studies, our use of the statistical measures of means and standard deviations of the key presses and transitions is uncommon and only practical for long text input. As additional features, we use percentages of key presses of many of the special keys. Some of these percentage features are designed to capture the user's preferences for using certain keys or key groups – for example, some users do not capitalize or use much punctuation. Other percentage features are designed to capture the user's pattern of editing text since there are many ways to locate (using keys – Home, End, Arrow keys – or mouse clicks), delete (Backspace or Delete keys, or Edit-Delete), insert (Insert, shortcut keys, or Edit-Paste), and move (shortcut keys or Edit-Cut, Edit-Paste) words and characters.

For this study, the feature vector consists of the 239 measurements listed in Table 1, which are also depicted in Figures 4 and 5. These features make use of the letter and digraph frequencies in English text [6], and the definitions of left-hand-letter keys as those normally struck by fingers of a typist's left hand (q, w, e, r, t, a, s, d, f, g, z, x, c, v, b) and right-hand-letter keys as those struck by fingers of the right hand (y, u, i, o, p, h, j, k, l, n, m). The features characterize a 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 mouse clicks, and the typing speed. The 239 features are grouped as follows:

- 78 duration features (39 means and 39 standard deviations) of individual letter and non-letter keys, and of groups of letter and non-letter keys,
- 70 type-1 transition features (35 means and 35 standard deviations) of the transitions between letters or groups of letters, between letters and non-letters or groups thereof, between non-letters and letters or groups thereof, and between non-letters and non-letters or groups thereof,
- 70 type-2 transition features (35 means and 35 standard deviations) which are identical to the 70 type-1 transition features except for the use of the type-2 transition measurement,
- 19 percentage features that measure the percentage of use of the non-letter keys and mouse clicks,
- 2 keystroke input rates: the unadjusted input rate (total time to enter the text / total number of keystrokes and mouse events) and the adjusted input rate (total time to enter the text minus pauses greater than ½ second / total number of keystrokes and mouse events).

#	Meas	Feature Measured	#	Meas	Feature Measured
1-2	μ & σ	dur all keystrokes	131-32	μ & σ	tran1 letter/non-letter
3-4	μ & σ	dur all alphabet letters	133-34	μ & σ	tran1 letter/space
5-6	μ & σ	dur vowels	135-36	μ & σ	tran1 letter/punct
7-8	μ & σ	dur vowels a	137-38	μ & σ	tran1 non-letter/letter
9-10	μ & σ	dur vowels e	139-40	μ & σ	tran1 shift/letter
11-12	μ & σ	dur vowels i	141-42	μ & σ	tran1 space/letter
13-14	μ & σ	dur vowels o	143-44	μ & σ	tran1 non-letter/non-letter
15-16	μ & σ	dur vowels u	145-46	μ & σ	tran1 space/shift
17-18	μ & σ	dur freq cons	147-48	μ & σ	tran1 punct/space
19-20	μ & σ	dur freq cons t	149-50	μ & σ	tran2 any-key/any-key
21-22	μ & σ	dur freq cons n	151-52	μ & σ	tran2 letter/letter
23-24	μ & σ	dur freq cons s	153-54	μ & σ	tran2 top cons pairs
25-26	μ & σ	dur freq cons r	155-56	μ & σ	tran2 top cons pairs th
27-28	μ & σ	dur freq cons h	157-58	μ & σ	tran2 top cons pairs st
29-30	μ & σ	dur next freq cons	159-60	μ & σ	tran2 top cons pairs nd
31-32	μ & σ	dur next freq cons l	161-62	μ & σ	tran2 vowel/cons
33-34	μ & σ	dur next freq cons d	163-64	μ & σ	tran2 vowel/cons an
35-36	μ & σ	dur next freq cons c	165-66	μ & σ	tran2 vowel/cons in
37-38	μ & σ	dur next freq cons p	167-68	μ & σ	tran2 vowel/cons er
39-40	μ & σ	dur next freq cons f	169-70	μ & σ	tran2 vowel/cons es
41-42	μ & σ	dur least freq cons	171-72	μ & σ	tran2 vowel/cons on
43-44	μ & σ	dur least freq cons m	173-74	μ & σ	tran2 vowel/cons at
45-46	μ & σ	dur least freq cons w	175-76	μ & σ	tran2 vowel/cons en
47-48	μ & σ	dur least freq cons y	177-78	μ & σ	tran2 vowel/cons or
49-50	μ & σ	dur least freq cons b	179-80	μ & σ	tran2 cons/vowel
51-52	μ & σ	dur least freq cons g	181-82	μ & σ	tran2 cons/vowel he
53-54	μ & σ	dur least freq cons other	183-84	μ & σ	tran2 cons/vowel re
55-56	μ & σ	dur all left hand letters	185-86	μ & σ	tran2 cons/vowel ti
57-58	μ & σ	dur all right hand letters	187-88	μ & σ	tran2 vowel/vowel
59-60	μ & σ	dur non-letters	189-90	μ & σ	tran2 vowel/vowel ea
61-62	μ & σ	dur space	191-92	μ & σ	tran2 double letters
63-64	μ & σ	dur shift	193-94	μ & σ	tran2 left/left
65-66	μ & σ	dur punctuation	195-96	μ & σ	tran2 left/right
67-68	μ & σ	dur punctuation .period	197-98	μ & σ	tran2 right/left
69-70	μ & σ	dur punctuation ,comma	199-200	μ & σ	tran right/right
71-72	μ & σ	dur punctuation 'apost	201-02	μ & σ	tran2 letter/non-letter
73-74	μ & σ	dur punctuation other	203-04	μ & σ	tran2 letter/space
75-76	μ & σ	dur numbers	205-06	μ & σ	tran2 letter/punct
77-78	μ & σ	dur other	2070-8	μ & σ	tran2 non-letter/letter
79-80	μ & σ	tran1 any-key/any-key	209-10	μ & σ	tran2 shift/letter
81-82	μ & σ	tran1 letter/letter	211-12	μ & σ	tran2 space/letter
83-84	μ & σ	tran1 top cons pairs	213-14	μ & σ	tran2 non-letter/non-letter
85-86	μ & σ	tran1 top cons pairs th	215-16	μ & σ	tran2 space/shift
87-88	μ & σ	tran1 top cons pairs st	217-18	μ & σ	tran2 punct/space
89-90	μ & σ	tran1 top cons pairs nd	219	%	shift
91-92	μ & σ	tran1 vowel/cons	220	%	caps lock
93-94	μ & σ	tran1 vowel/cons an	221	%	space
95-96	μ & σ	tran1 vowel/cons in	222	%	backspace
97-98	μ & σ	tran1 vowel/cons er	223	%	delete
99-100	μ & σ	tran1 vowel/cons es	224	%	insert
101-02	μ & σ	tran1 vowel/cons on	225	%	home
103-04	μ & σ	tran1 vowel/cons at	226	%	end
105-06	μ & σ	tran1 vowel/cons en	227	%	enter
107-08	μ & σ	tran1 vowel/cons or	228	%	ctl
109-10	μ & σ	tran1 cons/vowel	229	%	four arrow keys combined
111-12	μ & σ	tran1 cons/vowel he	230	%	Sent ending punct.?!
113-14	μ & σ	tran1 cons/vowel re	231	%	Other punct
115-16	μ & σ	tran1 cons/vowel ti	232	%	left shift
117-18	μ & σ	tran1 vowel/vowel	233	%	right shift
119-20	μ & σ	tran1 vowel/vowel ea	234	%	left mouse click
121-22	μ & σ	tran1 double letters	235	%	right mouse click
123-24	μ & σ	tran1 left/left	236	%	double left mouse click
125-26	μ & σ	tran1 left/right	237	%	left shift to right shift
127-28	μ & σ	tran1 right/left	238	rate	input rate with pauses
129-30	μ & σ	tran1 right/right	239	rate	input rate w/o pauses

Table 1: Summary of the 239 features used in this study.

The granularity of the duration and transition features is shown in the hierarchy trees of Figures 4 and 5. For each of these trees, the granularity increases from gross features at the top of the tree to fine features at the bottom. The least frequent letter in the duration tree is "g" with a frequency of 1.6%, and the least frequent letter pair in the transition tree is "or" with a frequency of 1.1% [6]. Because these features were designed to capture the keystroke patterns of users creating emails of as few as 200 keystrokes, we omit the infrequent alphabet letters, letter pairs, and punctuation, as well as the individual number keys and other infrequently used keys.

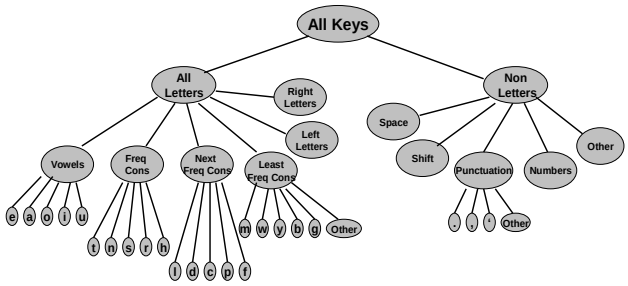

Figure 4: Hierarchy tree for the 39 duration categories (each oval), each represented by a mean and a standard deviation.

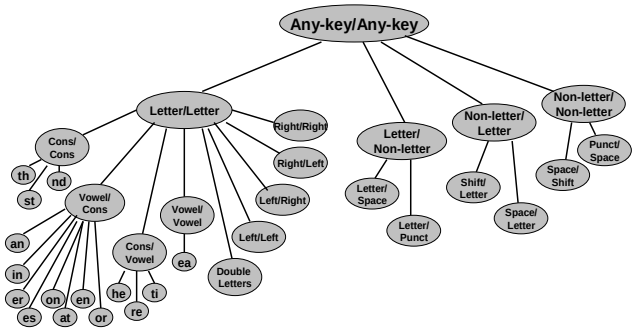

Figure 5: Hierarchy tree for the 35 transition categories (each oval), each represented by a mean and a standard deviation for each of the type 1 and type 2 transitions.

The computation of a keystroke-duration mean (μ) or standard deviation (σ) requires special handling when there are few samples. For this we use a fallback procedure which is similar to the "*backoff*" procedures used in natural language processing [11]. To compute μ for few samples – that is, when the number of samples is less than $k_{fallback\text{-}threshold}$ (an experimentally-optimized constant) – we take the weighted average of μ of the key in question and μ of the appropriate fallback as follows:

$$\mu'(i) = \frac{n(i) \cdot \mu(i) + k_{fallback-weight} \cdot \mu(fallback)}{n(i) + k_{fallback-weight}} \quad (1)$$

where $\mu'(i)$ is the revised mean, $n(i)$ is the number of occurrences of key i, $\mu(i)$ is the mean of the $n(i)$ samples of key i, $\mu(fallback)$ is the mean of the fallback, and $k_{fallback\text{-}weight}$ is the weight (an experimentally-optimized constant) applied to the fallback statistic. The appropriate fallback is determined by the next highest node in the hierarchy tree. For example, the "e" falls back to "vowels," which falls back to "all letters," which falls back to "all keys." The $\sigma(i)$ are similarly computed, as are the means and standard deviations of the transitions. Thus, we ensure the computability (no zero divides) and obtain reasonable values for all feature measurements.

Two preprocessing steps are performed on the feature measurements, outlier removal and feature standardization. Outlier removal consists of removing any duration or transition time that is far (more than $k_{outlier\text{-}\sigma}$ standard deviations) from the subject's $\mu(i)$ or $\mu(i, j)$, respectively. After outlier removal, averages and standard deviations are recalculated. The system can perform outlier removal a fixed number of times, recursively, or not at all, and this parameter, $k_{outlier\text{-}pass}$, is experimentally optimized. Outlier removal is particularly important for these features because a keyboard user could pause for a phone call, for a sip of coffee, or for numerous other reasons, and the resulting outliers (usually overly long transition times) could skew the feature measurements. Using a hill-climbing method, the four parameters – $k_{fallback\text{-}threshold}$, $k_{fallback\text{-}weight}$, $k_{outlier\text{-}\sigma}$, and $k_{outlier\text{-}pass}$ – were optimized on data from an earlier study [1].

After performing outlier removal and recalculation, we standardize the measurements by converting raw measurement x to x' by the formula,

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad (2)$$

where min and max are the minimum and maximum of the measurement over all samples from all subjects [5]. This provides measurement values in the range 0-1 to give each measurement roughly equal weight.

2.3. Classification

A Nearest Neighbor classifier, using Euclidean distance, compares the feature vector of the test sample in question against those of the samples in the training (enrollment) set. The author of the training sample having the smallest Euclidean distance to the test sample is identified as the author of the test sample.

3. Experiments

3.1 Subjects and Data Collection

After discarding incomplete data, a total of 118 subjects participated in the experiments. Data samples were obtained from students in introductory computer classes (accounting for the majority of the data samples); from students in classes at the masters and doctoral levels; and from friends, family, work colleagues, and fellow academics. Table 2 summarizes the demographic information of the subject pool.

Age	Female	Male	Total
Under 20	15	19	**34**
20-29	12	23	**35**
30-39	5	10	**15**
40-49	7	11	**18**
50-59	11	4	**15**
60+	0	1	**1**
All	**50**	**68**	**118**

Table 2: Summary of subject demographics

3.2 Experimental Design

Experiments were designed to explore the effectiveness of identifying users under optimal (same keyboard type and input mode for enrollment and testing) and non-optimal conditions (different type of keyboard, different mode of input, or both, for enrollment and testing). All the desktop keyboards were manufactured by Dell (and the data obtained primarily in classroom environments); over 70% of the laptop keyboards (mostly individually owned) were also by Dell, and the remaining ones were a mix of IBM, Compaq, Apple, HP, and Toshiba keyboards. We used two input modes: a copy-task in which subjects copied a predefined text of approximately 650 keystrokes, and free-text input in which subjects typed arbitrary emails of at least the length of the copy passage.

Figure 6 summarizes the experimental design, and shows four quadrants and six arrows. The quadrants are the areas in which the subjects were asked to participate: desktop copy, laptop copy, desktop free text, and laptop free text. The six arrows correspond to six experimental groupings. Groups 1 and 2 compare the two keyboard types on the copy-task and free-text inputs, respectively. Groups 3 and 4 compare the two input modes on the desktop and laptop keyboards, respectively. Finally, groups 5 and 6 compare the two possible ways of having different keyboard types and input modes for enrollment and testing. The subjects were asked to complete a minimum of two of the four quadrants as indicated by the two horizontal (1 and 2) and the two vertical (3 and 4) arrows in Figure 6. A subject completes a quadrant by typing a minimum of 5 samples of that category.

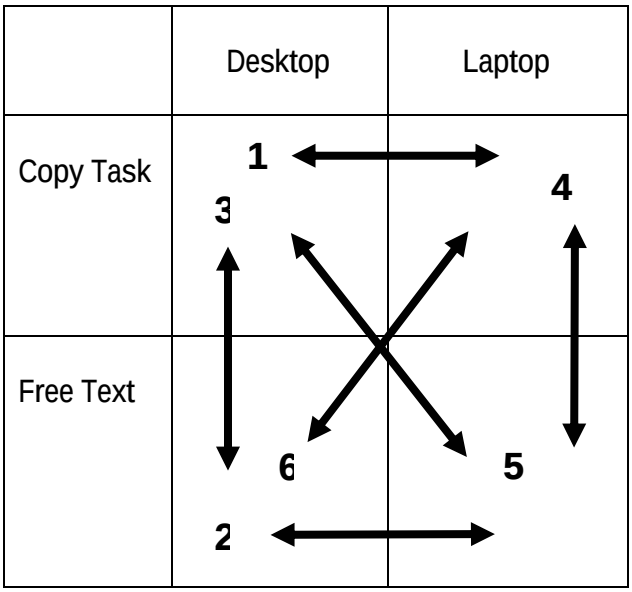

Figure 6: Summary of experimental design.

For the copy task on different keyboards (group/arrow 1), the subjects typed a copy of the predefined passage five times on one keyboard and then typed the same text five times on the other keyboard. For the free-text experiment on different keyboards (group/arrow 2), the subjects typed five arbitrary emails on each keyboard type. These two experimental groupings required the subjects to use both keyboard types. Groups 3 and 4 required the subjects to type in different modes on the same type of keyboard, so these groups were most suited for subjects having access to only one keyboard.

Although all subjects were invited to participate in all four quadrants of the experiment, due to time or equipment limitations some opted for two (minimum) while others participated in three or four quadrants of the experiment. This provided a comparison of different tasks on different keyboards as depicted by arrows 5 and 6 on the diagonals of Figure 6. A total of 118 subjects supplied five entries in at least two quadrants of the experiment (incomplete sample sets were discarded), and 36 completed all four quadrants of the experiment.

4. Results and Conclusions

The results of the study are summarized in Tables 2 and 3. Table 2 contains the results of the 36 users who completed all four quadrants of the experiments, and Table 3 contains the results of all the users who completed at lease two of the four quadrants. The six experimental groupings in each of these tables correspond to the six numbered arrows in Fig. 6. Several experiments were conducted within each experimental group, such as experiments 1-5 within group 1. Within each experimental group we tested under the optimal conditions of the same keyboard type and the same input mode (e.g., experiments 1 and 2 in group 1), the combined data (e.g., experiment 3 in group 1), and the less-optimal experimental conditions (e.g., experiments 4 and 5 in group 1). The combined data experiments combined the data from both quadrants covered by the arrow – from both the keyboard types, from both the input modes, or from both the keyboard types and the input modes. For the optimal conditions and for the combined experiments we used the leave-one-out classification procedure. For the less-optimal conditions we trained on one set and tested on the other. In Table 2 the two experiments for optimal conditions are only shown for groups 1 and 2 and need not be repeated for the remaining groups because the same 36 subjects participated in each experimental group, whereas this was not the case for the groups in Table 3.

The results in Table 2 are first discussed. The most important finding is that the system can identify with a high degree of accuracy the author of long-text input (either copy or free-text) as long as the author uses the same type of keyboard for both enrollment and testing, or when the data are combined. As anticipated, accuracy is highest under optimal and combined conditions (greater than 98%) when the population of users is relatively small, as in the 36-subject experiments.

Compared to the copy task, accuracy decreased somewhat for free-text input – from 100.0% to 99.5% on laptop keyboards and from 99.4% to 98.3% on desktop keyboards. This is perhaps understandable since each free-text sample was a different text whereas the copy samples were the same text. Interestingly, other variables being equal, the laptop accuracies were higher than the desktop accuracies – 100.0% versus 99.4% for the copy task and 99.5% versus 98.3% for free-text input. The reason for this might be the greater variety of laptop keyboards used in the experiments and the subject's greater familiarity with the laptop keyboards since the laptops were usually owned by the subjects.

Accuracy decreased significantly when the subjects used the same copy or free-text input mode but different keyboard types for enrollment and testing (experiments 4-5, and 9-10). For the copy task, the accuracy decrease in going from desktop/desktop (experiment 1) to desktop/laptop (experiment 4) was highly significant ($p = 2.0E-20$ using the Chi-square test). However, the difference between laptop/laptop and desktop/desktop (experiments 1 and 2) was not significant ($p = .31$). Finally, there was no significant difference ($p = .30$) between the copy task and free text on the desktop keyboard (experiments 1 and 6).

Accuracy also decreased significantly when the subjects used the same keyboard type but different input modes (experiments 12-13, and 15-16). The decrease in accuracy for different input modes was not as great as for

different keyboard types. The difference in accuracy for the different input modes suggests that an individual's keystroke patterns are significantly different for the two modes. Accuracy decreased most significantly when the subjects used different keyboard types and different input modes (experiments 18-19 and 21-22).

The results in Table 4 with the larger number of subjects support the above conclusions and also quantify the decrease in accuracy as the population of users increases.

Experiment	#	Train	Test	Accuracy
1. Copy Task *(36 subjects)*	1	Desktop	Desktop	99.4%
	2	Laptop	Laptop	100.0%
	3	Combined Keyboards	Combined Keyboards	99.5%
	4	Desktop	Laptop	60.8%
	5	Laptop	Desktop	60.6%
2. Free Text *(36 subjects)*	6	Desktop	Desktop	98.3%
	7	Laptop	Laptop	99.5%
	8	Combined Keyboards	Combined Keyboards	98.1%
	9	Desktop	Laptop	59.0%
	10	Laptop	Desktop	61.0%
3. Desktop *(36 subjects)*	11	Combined Modes	Combined Modes	99.2%
	12	Copy	Free Text	89.3%
	13	Free Text	Copy	91.7%
4. Laptop *(36 subjects)*	14	Combined Modes	Combined Modes	98.9%
	15	Copy	Free Text	86.2%
	16	Free Text	Copy	91.0%
5. Different Mode/Keyboard *(36 subjects)*	17	Combined Keyboards & Modes	Combined Keyboards & Modes	98.6%
	18	Desk Copy	Lap Free	51.6%
	19	Lap Free	Desk Copy	58.0%
6. Different Keyboard/Mode *(36 subjects)*	20	Combined Keyboards & Modes	Combined Keyboards & Modes	98.9%
	21	Lap Copy	Desk Free	50.3%
	22	Desk Free	Lap Copy	52.1%

Table 3: Summary of results for the 36 subjects participating in all four quadrants of the experiment.

Experiment	#	Train	Test	Accuracy
1. Copy Task *(52 subjects)*	1	Desktop	Desktop	99.2%
	2	Laptop	Laptop	99.6%
	3	Combined	Combined	98.9%
	4	Desktop	Laptop	54.6%
	5	Laptop	Desktop	51.9%
2. Free Text *(40 subjects)*	6	Desktop	Desktop	96.4%
	7	Laptop	Laptop	98.1%
	8	Combined	Combined	97.3%
	9	Desktop	Laptop	59.1%
	10	Laptop	Desktop	62.4%
3. Desktop *(93 subjects)*	11	Copy	Copy	99.1%
	12	Free Text	Free Text	93.3%
	13	Combined	Combined	95.9%
	14	Copy	Free Text	73.7%
	15	Free Text	Copy	81.1%
4. Laptop *(47 subjects)*	16	Copy	Copy	99.2%
	17	Free Text	Free Text	97.9%
	18	Combined	Combined	98.6%
	19	Copy	Free Text	80.2%
	20	Free Text	Copy	87.7%
5. Different Mode/ Keyboard *(41 subjects)*	21	Desk Copy	Desk Copy	99.0%
	22	Lap Free	Lap Free	99.1%
	23	Combined	Combined	98.6%
	24	Desk Copy	Lap Free	51.37%
	25	Lap Free	Desk Copy	51.44%
6. Different Keyboard/Mode *(40 subjects)*	26	Lap Copy	Lap Copy	98.5%
	27	Desk Free	Desk Free	99.5%
	28	Combined	Combined	98.8%
	29	Lap Copy	Desk Free	44.2%
	30	Desk Free	Lap Copy	51.4%

Table 4: Summary of results for all subjects participating in a minimum of two quadrants of the experiment.

In order to check the sufficiency of the number of enrollment samples, the accuracy as a function of the number of enrollment samples was measured on the free-text desktop data from 93 subjects (Figure 7). Since each subject supplied five data samples per quadrant, the leave-on-out procedure left a maximum of four enrollment samples to match against for a correct response. Since accuracy remains relatively high after two enrollment samples are available, it appears that a small number of enrollment samples is sufficient for this application.

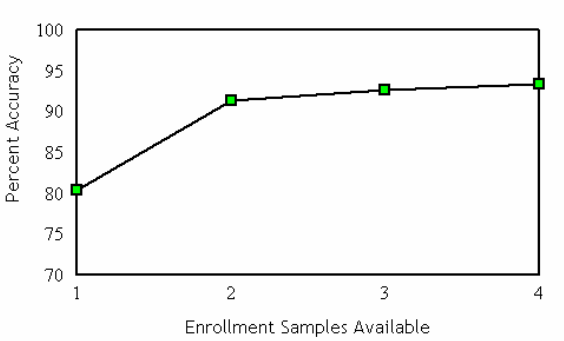

Figure 7: Accuracy versus enrollment samples (93 users).

To show the value of using the fallback procedure and outlier removal, we obtained results on the 93 subject free-text experiment under various parameter settings. Using the fallback procedure increased accuracy from 91.0% without fallback to 93.3% with fallback. Outlier removal was revisited (Figure 8) to demonstrate the importance of outlier removal. The setting used for outlier removal in the above experiments was "recursive" and the figure shows that about four passes of outlier removal are sufficient.

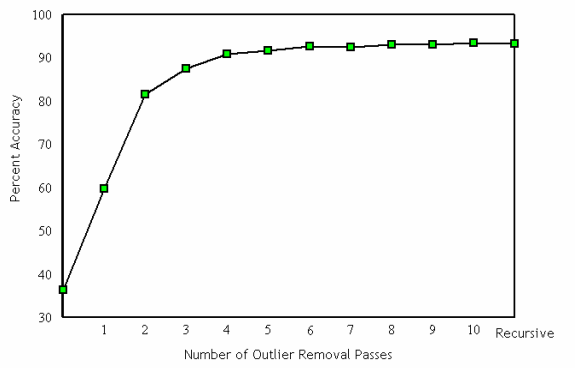

Figure 8: Accuracy versus number of outlier removal passes.

In summary, these results indicate that the keystroke biometric can be useful for identifying an individual who sends inappropriate email (free text) if sufficient enrollment samples are available and if the same type of keyboard is used to produce the enrollment and questioned samples. A secondary finding is that the keystroke biometric is significantly weaker for identification purposes when enrollment and testing are on different keyboard types. Finally, it is significant that accuracy is high on the free text mode (same keyboard) because the free text mode is of primary interest for the targeted application of identifying the author of inappropriate email.

Future work might explore the use of more sophisticated classifiers, such as Support Vector Machines (SVM). Also, although it is difficult to imagine how one could mimic another person's keystroke pattern, imposter performance might be investigated.

References

[1] Bartolacci, M. Curtin, M. Katzenberg, N. Nwana, S. Cha, and C.C. Tappert, "Applying Keystroke Biometrics for User Verification and Identification," *Proc. MCSCE*, MLMTA, Las Vegas, NV, June 2005.

[2] Bolle, R., Connell, J., Pankanti, S., Ratha, N., Senior, A.: Guide to Biometrics. Springer-Verlag, Berlin Heidelberg New York (2004).

[3] Brown, M., Rogers, S.J.: User Identification via Keystroke Characteristics of Typed Names using Neural Networks: International Journal of Man-Machine Studies, Vol. 39 No. 6 (1993) 999-1014.

[4] D'Souza, D.: Typing Dynamics Biometric Authentication. http://innovexpo.itee.uq.edu.au/2002/projects/s373901/thesis.PDF (2002) Accessed 5/05.

[5] Dunn, G. and Everitt, B.S.: An Introduction to Mathematical Taxonomy. Dover, New York (2004).

[6]Gaines, H.F.: Cryptanalysis: A Study of Ciphers and Their Solution. Dover, New York (1956), http://www-math.cudenver.edu/~wcherowi/courses/m5410/engstat.html, Accessed 1/06.

[7]Gaines R., Lisowski, W., Press, S., Shapiro, N.: Authentication by keystroke timing: some preliminary results: Rand Report R-256-NSF. Rand Corporation (1980).

[8]Gunnetti, D., Picardi, C.: Keystroke Analysis of Free Text: ACM, Vol. 8, No. 3 (2005) 312 – 347.

[9] IOSoftware: Authentication Basics IOSoftware. http://www.iosoftware.com/pages/Support/Authentication%20Basics/selection%20Process/index.asp (2005) Accessed 5/05.

[10] Jin, L., Ke, X., Manuel, R., Wilkerson, M.: Keystroke Dynamics: A Software based Biometric Solution: 13th USENIX Security Symposium (2004) .

[11] Jurafsky, D. and Martin, J.H., *Speech and Language Processing*, Prentice (2000).

[12] Leggett, J., Williams, G.: Verifying identity via keystroke characteristics: Int. J. Man-Machine Studies (1988) 67-76.

[13] Leggett, J., Williams, G., Usnick, M., Longnecker, M.: Dynamic identity verification via keystroke characteristics: Int J Man Machine Study (1991) 859-70.

[14] Monrose, F., Rubin, A.: Keystroke dynamics as a biometric for authentication: Future Generation Computer Systems, http://avirubin.com/fgcs.pdf (1999) Accessed 09/05.

[15] Obaidat, M.S., Sadoun, B.: Keystroke Dynamics Based Authentication http://web.cse.msu.edu/~cse891/Sect601/textbook/10.pdf (1998) Accessed 6/05.

[16] Peacock, A.: Learning User Keystroke Latency Patterns, http://pel.cs.byu.edu/~alen/personal/Coursework/cs572/KeystrokePaper/index.html, (2000) Accessed 5/05.

[17] Peacock, A., Ke, X., Wilerson, M.: Typing Patterns: A Key to User Identification: IEEE, Vol 2, No. 5 (2005) 40-47.

[18] Woodward, J. Jr., Orlans, N., Higgins, P., In: Biometrics: McGraw-Hill, New York (2003) 107.

[19] Yu, E., Cho, S.: Keystroke dynamics identity verification – its problems and practical solutions: http://dmlab.snu.ac.kr/ResearchPapers/Kestrokedynamicsidentityverification_itsproblemsandpracticalsolutions.pdf (2003) Accessed 5/05.