Keystroke Biometric Identification and Authentication on Long-Text Input

Charles C. Tappert, Mary Villani, and Sung-Hyuk Cha
Seidenberg School of CSIS, Pace University, Pleasantville, New York, 10570, USA

Abstract

A keystroke biometric system for long-text input was developed and evaluated for identification and authentication applications. The system consists of a Java applet to collect raw keystroke data over the Internet, a feature extractor, and pattern classifiers to make identification or authentication decisions. Experiments on over 100 subjects investigated two input modes – copy and free-text input – and two keyboard types – desktop and laptop keyboards. The system can accurately identify or authenticate individuals if sufficient enrollment samples are available and if the same type of keyboard is used to produce the enrollment and questioned input samples. Identification and authentication performance decreased significantly when subjects used different input modes or different keyboard types for enrollment and testing. Longitudinal experiments quantified performance degradation over intervals of several weeks and over an interval of two years. Additional experiments investigated the system's hierarchical model, parameter settings, assumptions, and sufficiency of enrollment samples and input-text length.

Keywords: biometrics, pattern recognition, behavioral biometrics, keystroke biometric, user authentication, user identification

Introduction

This chapter concerns identification and authentication applications of the keystroke biometric for long-text input of about 650 keystrokes, which is a short paragraph of about eight lines. An example identification application is a small 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. An authentication application is verifying the identity of students taking online quizzes or tests, which is an application becoming more important with the student population of online classes increasing and instructors becoming concerned about evaluation security and academic integrity. Finally, with more businesses moving to e-commerce, the keystroke biometric in Internet applications can provide an effective balance between high security and customer ease-of-use (Yu & Cho, 2004).

The keystroke biometric is one of the less-studied behavioral biometrics. Keystroke biometric systems measure typing characteristics believed to be unique to an individual and difficult to duplicate (Bolle, Connell, Pankanti, Ratha, & Senior, 2004; Jin, Ke, Manuel, & Wilkerson, 2004). Although most of the systems developed have been experimental in nature, there is a commercial product, BioPassword, currently used for hardening passwords (short input) in existing computer security schemes (Obiadat & Sadoun, 1999).

The keystroke biometric is appealing for several reasons. First, it is not intrusive and computer users type frequently for both work and pleasure. Second, it is inexpensive since the only hardware required is a computer with keyboard. 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 (Gunetti & Picardi, 2005). This continuing verification throughout a computer session is sometimes referred to as dynamic verification (Leggett & Williams, 2005; Leggett, Williams, Usnick, & Longnecker, 1991).

Most of the previous work on the keystroke biometric has dealt with user authentication, and while some studies used long-text input (Bergadano, Gunetti, & Picardi, 2002; Gunetti & Picardi, 2005; Leggett & Williams, 2005), most used passwords or short name strings (Bolle et al., 2004; Brown & Rogers, 1993; Obaidat & Sadoun, 1999). Fewer studies have dealt with user identification (Gunetti & Picardi, 2005; Peacock, Ke, & Wilkerson, 2004; Song, Venable, & Perrig, 1997). Gunetti and Picardi (2005) focused on long free-text passages, similar to this research, and also attempted the detection of uncharacteristic patterns due to fatigue, distraction, stress, or other factors. Song et al. (1997) touched on the idea of detecting a change in identity through continuous monitoring.

Researchers tend to collect their own data and no known studies have compared techniques on a common database. Nevertheless, the published literature is optimistic about the potential of keystroke dynamics to benefit computer system security and usability (Woodward, Orlans, & Higgins, 2002). Gunetti and Picardi (2005) 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, using users' normal interactions with computers, *free text*, as we do in this research.

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 (Woodward et al., 2002). 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 only a small amount of information. Moreover, this information tends to vary for different keyboards, different environmental conditions, and different entered texts (Gunetti & Picardi, 2005).

The keystroke biometric system reported here is unique in several respects. First, it collects raw keystroke data over the Internet, which is desirable for Internet security applications such as those described above. Second, it focuses on long-text input where sufficient keystroke data are available to permit the use of powerful statistical feature measurements – and the number, variety, and strength of the measurements used in the system are much greater than those used by earlier systems reported in the literature. Third, it focuses on applications using arbitrary text input because copy texts are unacceptable for most applications of interest. However, because of the statistical nature of the features and the use of arbitrary text input, special statistical fallback procedures were incorporated into the system to handle the paucity of data from infrequently used keyboard keys.

This chapter extends two previous studies on the *identification* application of a long-text keystroke biometric system. The first previous study showed the feasibility of an earlier version of the identification system on a text copy task (Curtin et al., 2006). The second showed the effectiveness of an improved system under ideal conditions of a fixed text and keyboard, and

under less favorable conditions of arbitrary texts and different keyboard types for enrollment and testing (Villani et al., 2006). This chapter extends the earlier studies, essentially the second one, in several ways. First, it presents the results of the second study in a clearer manner. Second, it extends the system to include an authentication component and presents authentication results to complement the earlier identification results. Third, it collects new data and performs longitudinal studies on data collected over intervals of several weeks and over an interval of two years to quantify performance degradation over time. Finally, it conducts additional experiments to investigate the hierarchical model, the parameter settings, the normal distribution assumption for the primary feature measurements, and the sufficiency of the number of enrollment samples and input text length.

The remainder of the chapter is organized as follows. The next section describes the keystroke biometric system, having components for data capture, feature extraction, and classification. Subsequent sections describes the experimental design and data collection; the experimental results on identification, on authentication, on the longitudinal studies, and on the system model and parameters; and finally the conclusions and suggestions for future work.

Keystroke Biometric System

The keystroke biometric system consists of four components: raw keystroke data capture, feature extraction, classification for identification, and classification for authentication.

Raw Keystroke Data Capture

A Java applet collects keystroke data over the Internet (Figure 1). The user is required to type in his/her name, but no data is captured on this entry. The submission number is automatically incremented after each sample submission, so the subject can immediately start typing the next sample. If the user is interrupted during data entry, the "Clear" button blanks all fields, except name and submission number, allowing the user to redo the current entry.

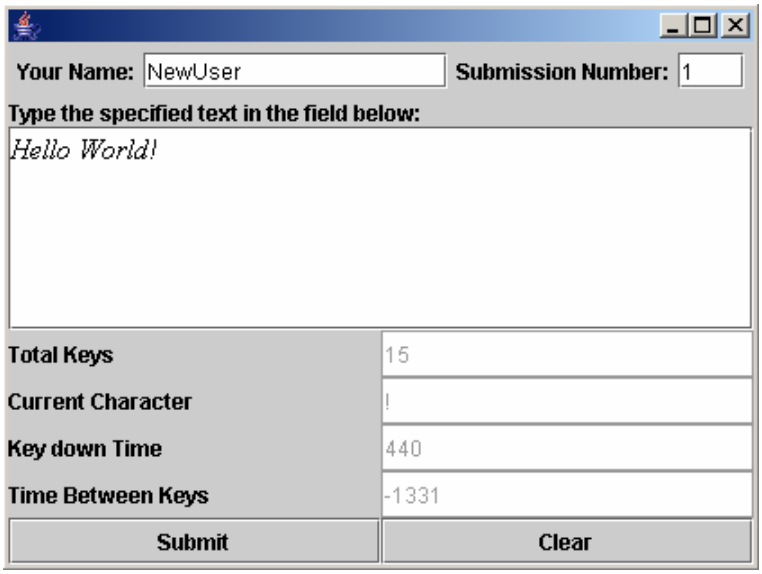


Figure 1. Java applet for data collection,
reprinted with permission from Villani et al. (2006).

Upon pressing submit, a raw-data text file is generated, which is delimited by the '~' character. Figure 2 shows the aligned version of the "Hello World!" raw data file. The raw data file contains the following information for each entry: 1) entry sequence number, 2) key's character, 3) key's code text equivalent, 4) key's location (1 = standard, only one key location; 2 = left side of keyboard; 3 = right side of keyboard), 5) time the key was pressed (milliseconds), 6) time the key was released (milliseconds). The number of left-mouse-click, right-mouse-click, and double left-mouse-click events during the session (these are events in contrast to key presses) are listed at the end of the file.

```
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!", reprinted with permission from Villani et al. (2006).

Feature Extraction

The system extracts a feature vector from the information in a raw data file. The features are statistical in nature and designed to characterize an individual's keystroke dynamics over writing samples of 200 or more characters. Most of the features are averages and standard deviations of key press duration times and of transition times between keystroke pairs, such as digraphs (Obaidat & Sadoun, 1999; Peacock et al., 2004). Figure 3 shows the transition between keystrokes measured in 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. While the second measure, t_2, is always positive because this sequence determines the keyboard output, the first measure, t_1, can be negative. We refer to these two measures of transition time as type-1 and type-2 transition features.

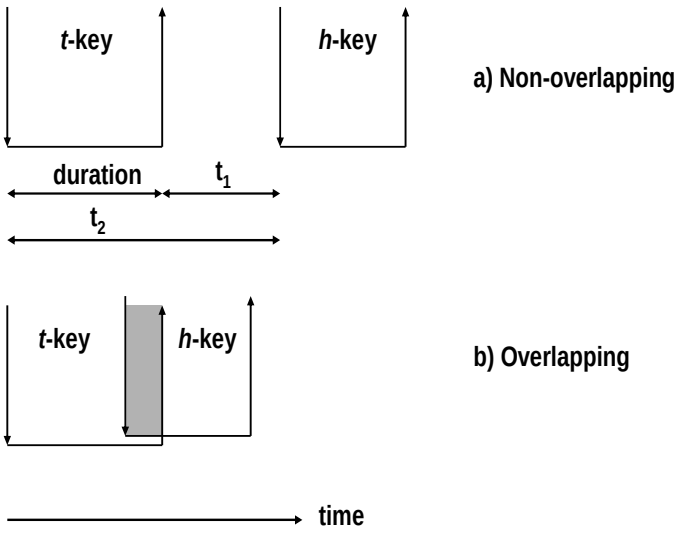


Figure 3. A two-key sequence (*th*) shows the two transition measures: t_1 = *press* time of second key – *release* time of first, and t_2 = *press* time of second key – *press* time of first. A keystroke is depicted as a bucket with the down arrow marking the press and the up arrow the release time. 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). Reprinted with permission from Villani et al. (2006).

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.

This study used 239 feature measurements (a complete list is presented in the Appendix). These features make use of the letter and digraph frequencies in English text (Gains, 1956), 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 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% (Gains, 1956). The six least frequent letters are grouped under "other" and the infrequent digraphs are also grouped. 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 (Figure 4)

- 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 (Figure 5)
- 70 type-2 transition features (35 means and 35 standard deviations) identical to the type-1 transition features except for the method of measurement (Figure 5)
- 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)

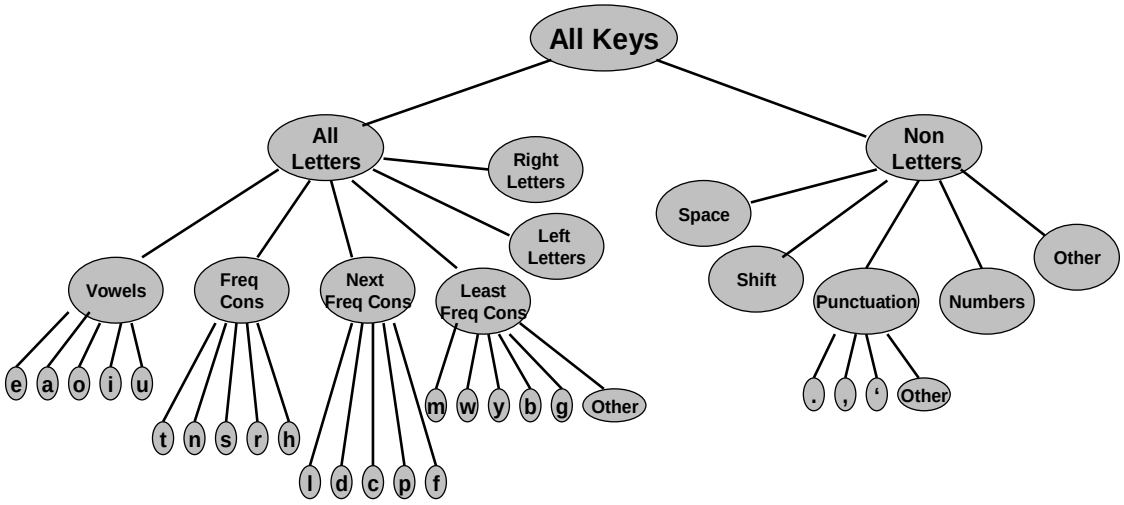


Figure 4. Hierarchy tree for the 39 duration categories (each oval), reprinted with permission from Villani et al. (2006).

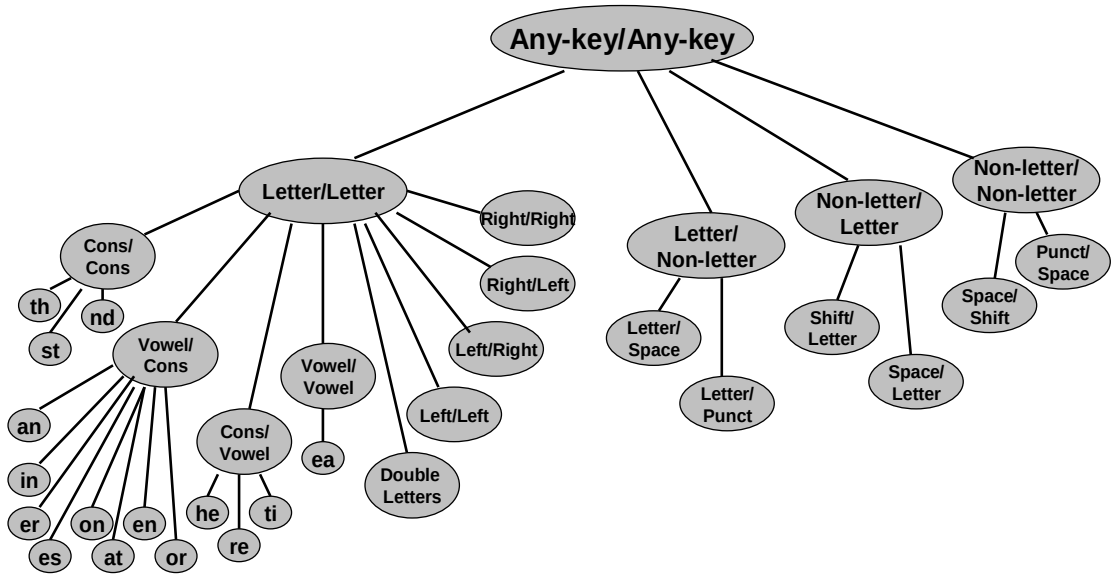


Figure 5. Hierarchy tree for the 35 transition categories (each oval) for type 1 and type 2 transitions, reprinted with permission from Villani et al. (2006).

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 (Jurafsky & Martin, 2000). 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:

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

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 "m" falls back to "least frequent consonants", which falls back to "all letters", which falls back to "all keys". Because we are dealing with long-text input, fallback is necessary for only infrequently used keys; thus, it is based primarily on frequency of use and fallback of more than one level is rare. 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 different data from an earlier study (Curtin et al., 2006).

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

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

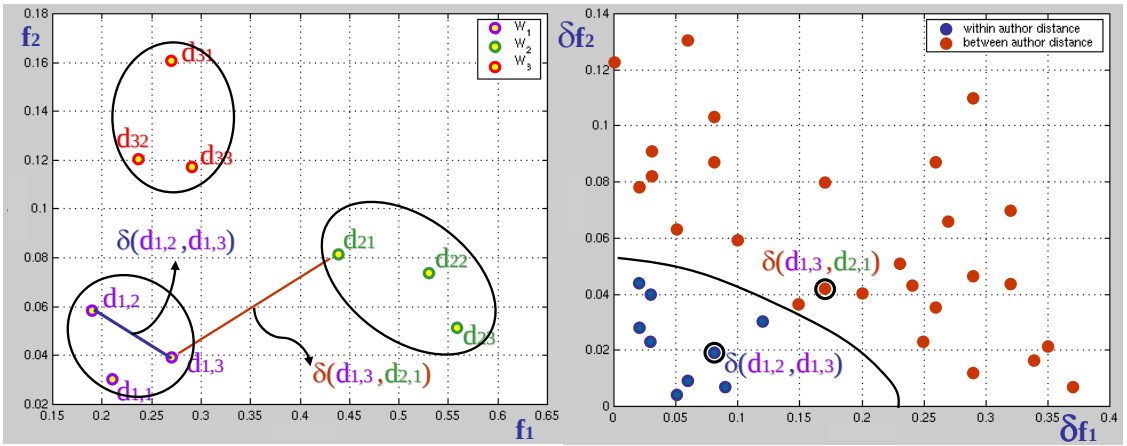
where min and max are the minimum and maximum of the measurement over all samples from all subjects (Dunn & Everitt , 2004). This provides measurement values in the range 0-1 to give each measurement roughly equal weight.

Classification for Identification

For identification, 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.

Classification for Authentication

For authentication, a vector-difference dichotomy model transforms a multi-class (polychotomy) problem into a two-class (dichotomy) problem (Figure 6) (Choi, Yoon, Cha, & Tappert , 2004; Yoon, Choi, Cha, Lee, & Tappert, 2005).



(a) Feature space (Polychotomy)

(b) Feature distance space (Dichotomy)

Figure 6. Authentication transformation from (a) Feature space to (b) Feature distance space, reprinted with permission from Yoon et al. (2005).

To explain the dichotomy transformation process, take an example of three people $\{P_1,P_2,P_3\}$ where each person supplies three biometric samples. Figure 6 (a) plots the biometric sample data for these three people in the feature space, exemplifying the polychotomy model. This

feature space is transformed into a distance vector space by calculating vector distances between pairs of samples of the *same* person (*intra-person distances*, denoted by $x_{\oplus}$) and distances between pairs of samples of *different* people (*inter-person distances*, denoted by $x_{\varnothing}$). Let d_{ij} represent the feature vector of the i^{th} person's j^{th} biometric sample, then $x_{\oplus}$ and $x_{\varnothing}$ are calculated as follows :

$$x_{\oplus} = |d_{ij} - d_{ik}| \text{ where } i=1 \text{ to } n, \text{ and } j,k=1 \text{ to } m, j \neq k \qquad (3)$$
$$x_{\varnothing} = |d_{ij} - d_{kl}| \text{ where } i,k=1 \text{ to } n, i \neq k \text{ and } j,l=1 \text{ to } m$$

where n is the number of people and m is the number of samples per person. Figure 6 (b) shows the transformed feature distance space for the example problem.

Yoon et al. (2005) derive the numbers of the inter- and intra-person distances. If n people provide m biometric samples each, the numbers of intra-person and inter-person distance samples, respectively, are:

$$n_{\oplus} = \frac{m \times (m-1) \times n}{2} \quad \text{and} \quad n_{\varnothing} = m \times m \times \frac{n \times (n-1)}{2} \qquad (4)$$

In the feature distance space we then use the Nearest Neighbor classifier, using Euclidean distance, to compare a feature vector distance against those in the training (enrollment) set. The training sample having the smallest Euclidean distance to the test sample is identified, and the test sample assigned as being intra-class (same person) or inter-class (different people) according the truth of that training sample.

Experimental Design and Data Collection

In this study, we vary two independent variables – keyboard type and input mode – to determine their effect on both identification and authentication performance. The keyboard types were desktop and laptop PC keyboards. The input modes were a copy task and free (arbitrary) text input. By varying these independent variables, we determined the distinctiveness of keystroke patterns when training and testing on long-text input under ideal conditions (same input mode and keyboard type for enrollment and testing) and under non-ideal conditions (different input mode, different type of keyboard, or both, for enrollment and testing).

All the desktop keyboards were manufactured by Dell and the data obtained primarily in classroom environments; over 90% of the smaller laptop keyboards (mostly individually owned) were also by Dell, and the others 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 652 keystrokes (515 characters with no spaces, 643 with spaces, and 652 including 9 shift-key presses for uppercase), and free-text input in which subjects typed arbitrary emails of at least 650 keystrokes. The subjects were instructed to correct errors, further increasing the number of keystrokes.

Figure 7 summarizes the experimental design and shows the subject pool. The two independent variables – the two keyboard types and the two input modes – yield four data quadrants. Data were collected in each quadrant: desktop copy, laptop copy, desktop free text, and laptop free text. There are four optimal (ideal) conditions – enrollment and testing on data within each of the four quadrants.

There are six non-optimal experimental groups corresponding to the six arrows in Figure 7 – training on data at one end of the arrow and testing on data at the other end (and since either end

of an arrow can be the starting point, there are a total of 12 non-optimal experimental conditions). Groups 1 and 2 compare the two input modes on the desktop and laptop keyboards, respectively. Groups 3 and 4 compare the two keyboard types on the copy-task and free-text inputs, respectively. Finally, groups 5 and 6 compare the two possible ways of having different keyboard types and different input modes for enrollment and testing. Note that although there are six experimental groups (arrows), there are three major experimental groupings – training and testing on different input modes (the two vertical arrows), different keyboard types (the two horizontal arrows), and both different input modes and different keyboard types (the two diagonal arrows).

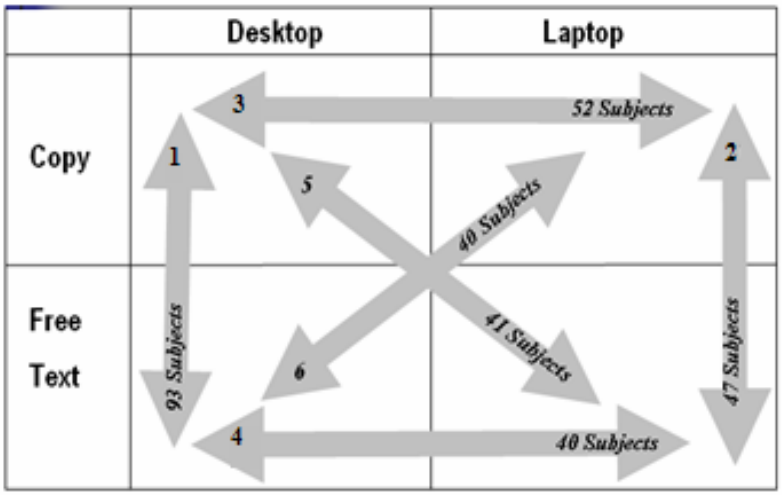


Figure 7. Experimental design showing the subject pool, adapted with permission from Villani (2006).

For data collection, the subjects were asked to complete a minimum of two of the four quadrants as indicated by the two horizontal and two vertical arrows in Figure 7. A subject completes a quadrant by typing a minimum of 5 samples of that category. 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.

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. 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 (Figure 7, Table 1).

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+	11	5	**16**
All	**50**	**68**	**118**

Table 1: Summary of subject demographics, adapted with permission from Villani (2006).

Data on the 118 subjects were collected in 2006. To collect reasonable amounts of data quickly the timing of the input samples was not controlled, and about half of the subjects input all their keystroke data samples in one sitting, while the others input their samples over several days or weeks. Similar data were collected in 2008 for the longitudinal studies, and the recording times of these data were accurately controlled.

For the copy and free-text tasks on a desktop keyboard (group 1), the subjects typed a copy of the predefined passage five times and then typed five arbitrary emails on a desktop keyboard. For the copy and free-text tasks on a laptop keyboard (group 2), the typing was similar but on a laptop keyboard. These two experimental groupings were most suited for subjects having access to only one keyboard. Groups 3 and 4 required the subjects to type in the same mode but on different keyboards. Finally, groups 5 and 6 required the subjects to type in different input modes and on different keyboard types.

Experimental Results

Experimental results are presented here for biometric identification, biometric authentication, longitudinal studies on both identification and authentication, and an investigation of the system hierarchical model and parameter settings.

Identification Experimental Results

The identification results of the study are summarized in Tables 2 and 3, and corresponding Figures 8 and 9, respectively. Table 2 and Figure 8 present the results under the *optimal conditions* of training (enrollment) and testing on data obtained using the same keyboard type and the same input mode. Since training and testing were under the same conditions, the leave-one-out procedure was used in order to test on data different from that used for training. As anticipated, performance (accuracy) is high under these optimal conditions – greater than 98% when the population of users is relatively small (36-subject experiment), and decreasing for larger numbers of subjects. This performance decrease as the number of subjects increases is highlighted in the average of the four cases at the bottom of Table 2, which indicates that doubling the number of subjects increases the error rate by about a factor of four (from 0.7% to 2.6%). The graphs of the four optimal-condition cases in Figure 8 also show the large effect of population increase on performance.

Conditions	**36-Subject**		**Full-Subject**	
Train and Test	**Subjects**	**Accuracy**	**Subjects**	**Accuracy**
DeskCopy	36	99.4%	93	99.1%
LapCopy	36	100.0%	47	99.2%
DeskFree	36	98.3%	93	93.3%
LapFree	36	99.5%	47	97.9%
Average	**36**	**99.3%**	**70**	**97.4%**

Table 2. Identification performance under optimal conditions.

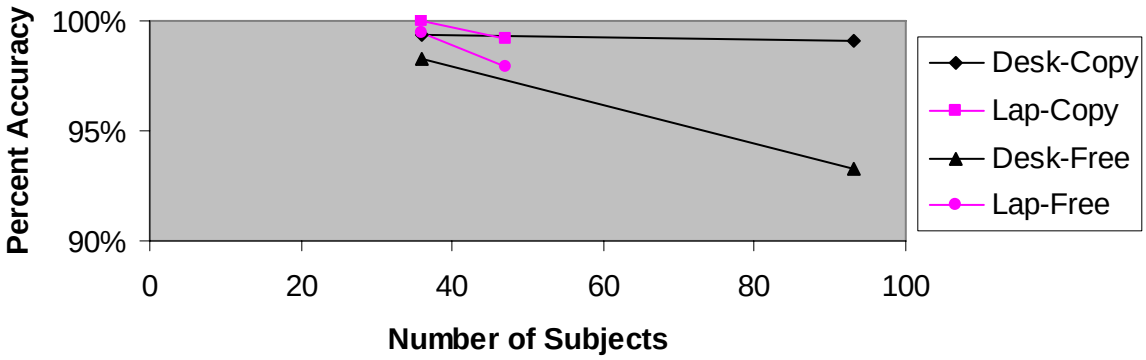


Figure 8. Identification performance under optimal conditions, graphs of results from Table 2.

Under optimal conditions in the 36-subject experiment, accuracy varied somewhat from quadrant to quadrant. For example, accuracy was a little higher on the copy task compared to free-text input – 99.4% compared to 98.3% on desktop keyboards, and 100.0% compared to 99.5% on laptop keyboards. These differences, however, were not statistically significant – for example, the first difference yielded a null hypothesis $p = 0.3$ (Chi-square was used for all tests of statistical significance). Higher copy task accuracy is understandable since the copy samples were of the same text whereas the free-text samples were of different texts. Also, other variables being equal, accuracy was a little higher on the laptop keyboards compared to the desktop keyboards – 100.0% compared to 99.4% for the copy task, and 99.5% compared to 98.3% for free-text input. These differences were also not statistically significant. The reason for higher laptop accuracy is likely the greater variety of laptop keyboards used in the experiments and the subject's greater familiarity with the laptops since they were usually owned by the subjects.

Table 3 and Figure 9 present the results under the *non-optimal conditions* of training and testing under different conditions – different input modes, different keyboard types, or both different input modes and different keyboard types. The graphs of Figure 9, which average the two cases in each of the six groups of Table 3, clearly show the degradation in performance as the conditions for training and testing go from different input modes (groups 1 and 2), to different keyboard types (groups 3 and 4), and finally to both different input modes and different keyboard types (groups 5 and 6). They also show the decrease in performance as the population increases.

Under non-optimal conditions in the 36-subject experiment, accuracy decreased from about 99% under optimal conditions to about 90% when the subjects used the same keyboard type but different input modes (the four cases in groups 1 and 2). For example the accuracy decrease in going from the optimal-condition DeskCopy/DeskCopy (99.4%) to the non-optimal- condition DeskCopy/DeskFree (89.3%) was statistically significant ($p < 0.0001$). Accuracy dropped even more significantly (from about 99% to about 60%) when the subjects used the same copy or free-text input mode but different keyboard types for enrollment and testing (groups 3 and 4). Finally, accuracy decreased most significantly, from about 99% to about 53%, when the subjects used different input modes and different keyboard types (groups 5 and 6). These results suggest that an individual's keystroke patterns differ for the different input modes and the different keyboard types, and differ more for different keyboard types than for different input modes. Figure 9 graphically shows the performance on the three major conditions of training and testing on different input modes (groups 1 and 2), different keyboard types (groups 3 and 4), and both

different input modes and different keyboard types (groups 5 and 6), as well as the performance decrease as the number of subjects increase.

Group	Conditions		36-Subject		Full-Subject	
	Train	Test	Subjects	Accuracy	Subjects	Accuracy
1	DeskCopy	DeskFree	36	89.3%	93	73.7%
	DeskFree	DeskCopy	36	91.7%	93	81.1%
2	LapCopy	LapFree	36	86.2%	47	80.2%
	LapFree	LapCopy	36	91.0%	47	87.7%
3	DeskCopy	LapCopy	36	60.8%	52	54.6%
	LapCopy	DeskCopy	36	60.6%	52	51.9%
4	DeskFree	LapFree	36	59.0%	40	59.1%
	LapFree	DeskFree	36	61.0%	40	62.4%
5	DeskCopy	LapFree	36	51.6%	41	51.4%
	LapFree	DeskCopy	36	58.0%	41	51.4%
6	DeskFree	LapCopy	36	52.1%	40	44.2%
	LapCopy	DeskFree	36	50.3%	40	51.4%

Table 3. Identification performance under non-optimal conditions.

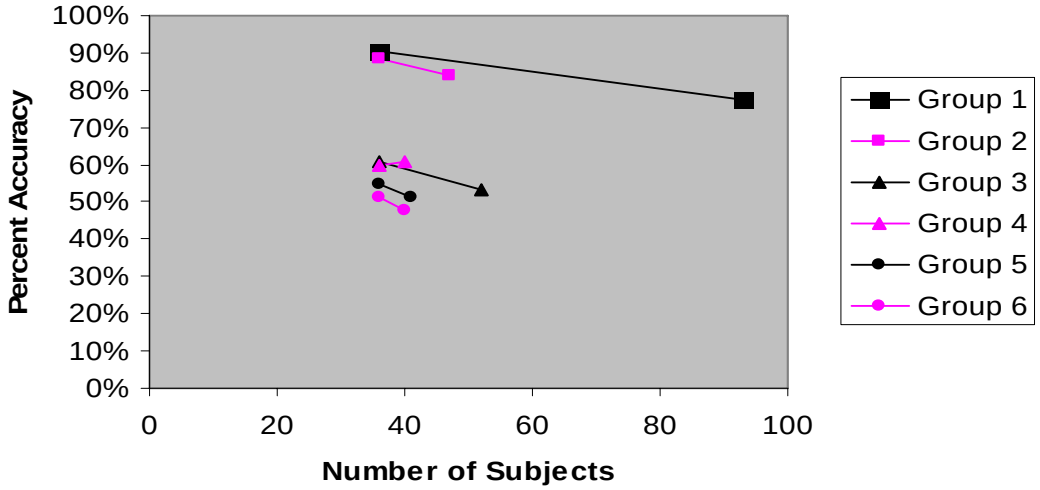


Figure 9. Identification performance under non-optimal conditions, graphs of averaged group results from Table 3.

Authentication Experimental Results

The authentication results are presented in Tables 4, 5, and 6. Tables 4 and 5 present the results under *optimal conditions* (same conditions for training and testing) on the 36-subject data, using 18 subjects for training and the remaining 18 for testing. The experiments in Table 4 used all the inter-class samples and those in Table 5 used a reduced set of inter-class samples. For the first test in Table 4, for example, the training and testing sets each consisted of 90 samples (18 subjects contributing 5 samples each), with all samples obtained under the

DeskCopy conditions. The intra- and inter-class sizes were 180 and 3825, respectively, and the tests in this case were run on all the dichotomy data without reducing the sample size of the inter class data (in the third test the smaller intra-inter class size is due to a few missing samples).

Conditions	Intra-Inter Class Sizes		FRR	FAR	Performance
	Train	Test			
DeskCopy	180-3825	180-3825	11.1%	6.0%	93.8%
LapCopy	180-3825	180-3825	7.8%	4.4%	95.5%
DeskFree	171-3570	176-3740	28.4%	1.4%	97.4%
LapFree	180-3825	180-3825	15.6%	3.7%	95.7%
Average			**15.7%**	**3.9%**	**95.6%**

Table 4. Authentication performance under optimal conditions, train 18 and test 18 different subjects using all inter-class samples.

Table 5 repeated the optimal-conditions experiments of Table 4 but used a reduced set of randomly selected 500 inter-class data samples. With fewer enrollment samples system performances decreased from roughly 95% to 90%, a doubling of the error rate, but FRR and FAR were closer in value because the numbers of intra and inter-class samples were more balanced.

Conditions	Intra-Inter Class Sizes		FRR	FAR	Performance
	Train	Test			
DeskCopy	180-500	180-500	10.0%	13.4%	87.5%
LapCopy	180-500	180-500	1.7%	10.2%	92.1%
DeskFree	171-500	176-500	18.8%	5.0%	91.4%
LapFree	180-500	180-500	9.4%	10.8%	89.6%
Average			**10.0%**	**9.9%**	**90.2%**

Table 5. Authentication performance under optimal conditions, train 18 and test 18 different subjects using 500 random inter-class samples.

Table 6 presents the results under *non-optimal conditions* (training on one condition and testing on another) on the 36-subject data. The tests were performed on a reduced set of 500 randomized inter-class samples because the full number of inter-class samples was over 15,000. Interestingly, the authentication results under non-optimal conditions only decreased from an average of 90.2% to 87.4% (Tables 5 and 6, respectively, on 500 inter-class samples), a small decrease compared to the corresponding decrease in the identification experiments, but partially explained by the larger number of intra-class samples under the non-optimal conditions. Furthermore, and somewhat surprisingly, the three different primary conditions showed essentially the same average performance – same keyboard type/different input modes (groups 1 and 2) an average performance of 87.0%, same input mode/different keyboard types (groups 3 and 4) 87.5%, and different input modes/different keyboard types (groups 5 and 6) 87.7%. Thus, although average performance dropped from 90.2% under optimal conditions to 87.4% under non-optimal conditions, the performance does not change significantly as we go from different input modes, to different keyboard types, and to different input modes and different keyboard types. We attribute these strong non-optimal results to the robustness of the authentication system.

	Conditions		Intra-Inter Class Sizes		FRR	FAR	Performance
	Train	Test	Train	Test			
1	DeskCopy	DeskFree	360-500	347-500	8.1%	17.8%	86.2%
	DeskFree	DeskCopy	347-500	360-500	3.3%	13.0%	91.0%
2	LapCopy	LapFree	360-500	360-500	3.6%	40.4%	75.0%
	LapFree	LapCopy	360-500	360-500	5.8%	3.4%	95.6%
3	DeskCopy	LapCopy	360-500	360-500	5.3%	6.8%	93.8%
	LapCopy	DeskCopy	360-500	360-500	4.7%	18.0%	87.6%
4	DeskFree	LapFree	347-500	360-500	3.1%	38.8%	76.2%
	LapFree	DeskFree	360-500	347-500	8.9%	6.8%	92.3%
5	DeskCopy	LapFree	360-500	360-500	5.8%	22.2%	84.7%
	LapFree	DeskCopy	360-500	360-500	5.3%	8.8%	92.7%
6	DeskFree	LapCopy	347-500	360-500	1.7%	14.4%	90.9%
	LapCopy	DeskFree	360-500	347-500	3.2%	27.6%	82.4%
Average					**4.9%**	**18.2**	**87.4**

Table 6. Authentication performance under non-optimal conditions, 36 subjects, using 500 random inter-class samples.

Longitudinal Study Results

In order to study the accuracy of identification and authentication over time, we performed studies at two-week intervals and at a two-year interval. The two-week interval study used 13 subjects who had not participated in the earlier experiments, and the two-year interval study brought back 8 of the participants of the earlier 36-subject study for additional data samples. All the longitudinal experimental results were obtained under *non-optimal conditions* – different keyboard type, different input mode, or both, for training (enrollment) and testing.

The identification and authentication results of the two-week-interval study are presented in the Tables 7 and 8, respectively. Baseline results were obtained by training and testing on data from the same week – week 0 (W0-W0), week 2 (W2-W2), and week 4 (W4-W4), and these three sets of results were combined to obtain overall "Same-Week" performance. For the two-week interval, results were obtained by training on week 0 and testing on week 2 (W0-W2) and by training on week 2 and testing on week 4 (W2-W4), and these two sets of results were combined for the overall "Two-Week Interval" performance. For the "Four-Week Interval", results were obtained by training on week 0 and testing on week 4 (W0-W4). Five samples were collected from each subject in each quadrant, for a total of 65 samples per file (with the exception of the week-4 laptop copy file, which was missing one sample for a total of 64 samples). Percentages shown are the percent of the samples correctly identified. The identification results (Table 7) shows the degree of performance degradation over time, summarized by the average performance (bottom line of table).

	Conditions		Same Week	Two Week Interval	Four Week Interval
	Train	Test			

1	DeskCopy	DeskFree	94.3	77.7	78.5
	DeskFree	DeskCopy	97.5	87.7	90.8
2	LapCopy	LapFree	94.3	93.1	90.8
	LapFree	LapCopy	91.7	89.1	89.1
3	DeskCopy	LapCopy	90.2	89.2	87.5
	LapCopy	DeskCopy	96.9	97.7	87.7
4	DeskFree	LapFree	85.7	79.2	76.9
	LapFree	DeskFree	96.4	91.6	87.7
5	DeskCopy	LapFree	81.7	74.2	70.7
	LapFree	DeskCopy	89.7	80.0	83.1
6	DeskFree	LapCopy	74.7	77.5	75.0
	LapCopy	DeskFree	85.1	86.2	81.5
Average Performance			**89.9**	**85.3**	**83.3**

Table 7. Identification performance on 13 subjects over two-week intervals.

The authentication results (Table 8) showed less performance degradation than the identification results over the two- and four-week intervals.

	Conditions		**Same Week**		**Two-Week Interval**		**Four-Week Interval**	
	Train	**Test**	**FAR/FRR**	**Perf.**	**FAR/FRR**	**Perf.**	**FAR/FRR**	**Perf.**
1	DeskCopy	DeskFree	2.8/4.1	96.1	3.0/4.3	95.9	3.8/4.4	95.7
	DeskFree	DeskCopy	3.9/7.6	92.9	3.1/10.8	90.1	3.1/16.1	85.4
2	LapCopy	LapFree	9.6/8.0	83.0	6.9/24.1	77.9	5.4/34.9	68.5
	LapFree	LapCopy	4.1/9.3	91.3	3.1/9.4	91.3	2.3/10.5	90.4
3	DeskCopy	LapCopy	2.1/3.3	96.8	2.3/4.2	96.1	6.9/1.8	97.6
	LapCopy	DeskCopy	6.7/25.1	77.0	5.8/33.4	69.8	4.6/35.6	68.0
4	DeskFree	LapFree	5.6/7.0	93.2	2.3/7.8	92.9	10.0/9.6	90.4
	LapFree	DeskFree	7.2/6.8	93.2	1.9/10.2	90.8	1.5/21.2	81.1
5	DeskCopy	LapFree	4.4/6.6	93.7	5.0/7.9	92.5	2.3/7.1	93.5
	Lap Free	DeskCopy	3.3/4.2	87.0	3.5/15.6	85.8	2.3/26.4	76.4
6	DeskFree	LapCopy	3.3/9.3	91.4	3.1/7.0	93.5	7.7/2.9	96.5
	LapCopy	DeskFree	6.5/9.7	81.8	5.8/30.1	72.7	7.1/21.6	80.0
Average Performance				**89.8**		**87.4**		**85.3**

Table 8. Authentication performance on 13 subjects over two-week intervals.

For the two-year interval study, we contacted each of the subjects who participated in the earlier 36-subject study in 2006 (Y0), and asked them to enter new complete data sets (5 samples in each of the four quadrants). New data sets were obtained from 8 of these individuals in 2008 (Y2), approximately two years after obtaining their earlier data. Since each of the 8 subjects submitted five samples in each of four quadrants, there were a total of 40 samples in each quadrant.

Both the Y0 and Y2 data from these 8 subjects were run through the system. The results of training and testing on data recorded in the same year, Y0-Y0 and Y2-Y2, were averaged. Table 9 shows the percent of the samples (80 samples in the "Same Year", half in Y0 and half in Y2; and 40 in the "Two-Year Interval") accurately identified. The resulting substantial degradation in performance indicates that one's keystroke patterns change significantly over a two-year interval.

Group	Conditions		Same Year	Two-Year Interval
	Train	Test		
1	DeskCopy	DeskFree	97.5	57.5
	DeskFree	DeskCopy	92.5	75.0
2	LapCopy	LapFree	98.8	60.0
	LapFree	LapCopy	100.0	57.5
3	DeskCopy	LapCopy	87.5	67.5
	LapCopy	DeskCopy	76.3	65.0
4	DeskFree	LapFree	80.0	80.0
	LapFree	DeskFree	76.3	80.0
5	DeskCopy	LapFree	80.0	65.0
	LapFree	DeskCopy	72.5	52.5
6	DeskFree	LapCopy	76.3	57.5
	LapCopy	DeskFree	75.0	82.5
Average Performance			**84.4**	**66.7**

Table 9. Identification performance on 8 subjects over a two-year interval.

Authentication results (Table 10) are better, with an average accuracy of 92% with a two-year interval between the training and test sets. Although this performance is better than that obtained over two- and four-week intervals, this is likely due to the smaller number of subjects in the two-year study.

	Conditions		Same Year		Two-Year Interval	
	Train	Test	FAR/FRR	Performance	FAR/FRR	Performance
1	DeskCopy	DeskFree	8.75 / 7.50	92.4	10.00/12.00	88.2
	DeskFree	DeskCopy	5.13 / 10.61	89.9	2.63 / 9.37	91.3
2	LapCopy	LapFree	0.00 / 2.86	97.4	0.00 / 3.71	96.7
	LapFree	LapCopy	1.25 / 3.29	96.9	0.00 / 7.28	93.5
3	DeskCopy	LapCopy	5.00 / 8.29	92.1	5.00 / 9.71	90.8
	LapCopy	DeskCopy	1.25 / 4.14	96.2	0.00 / 8.42	92.4
4	DeskFree	LapFree	0.64 / 8.33	92.4	3.94 / 5.82	94.4
	LapFree	DeskFree	1.56 / 2.21	97.7	2.50 / 3.42	96.7
5	DeskCopy	LapFree	5.63 / 10.29	90.2	6.25 / 12.85	87.8
	Lap Free	DeskCopy	3.13 / 8.86	91.7	2.50 / 9.00	91.7
6	DeskFree	LapCopy	0.00 / 10.40	90.6	1.31 / 13.49	87.7
	LapCopy	DeskFree	5.00 / 5.00	95.0	1.25 / 6.57	94.0
Average Performance				**93.5**		**92.1**

Table 10. Authentication performance on 8 subjects over a two-year interval.

System Hierarchical Model and Parameter Experiments

The hierarchical model was investigated and alternative models were evaluated. The system parameters were analyzed by measuring accuracy as a function of the outlier removal parameters (the number of outlier passes and the outlier distance), accuracy as a function of the number of enrollment samples, and accuracy as a function of input text length. The parameter experiments

were performed on the identification system using the full-subject optimal DeskFree condition, or both the DeskFree and DeskCopy conditions – the conditions having the largest number of 93 subjects. Finally, the normal distribution assumption of the statistical features was verified.

Hierarchical fallback model. We investigated the fallback aspect of the hierarchical model by comparing the hierarchical fallback as described above to simply falling back to the top nodes of the hierarchy trees as was done in an earlier study (Curtin et al., 2006). For the desktop-free condition the hierarchical fallback procedure increased accuracy from 91.0% to 93.3% (a 26% decrease in error rate). For the desktop-copy condition, identification accuracy increased from 98.1% to 99.1% (a 53% decrease in error rate). Using the hierarchical model for fallback is therefore highly beneficial.

An analysis of the fallback model showed that fallback never occurred more than one level up from the leaf nodes and that most of the one-level-up nodes were essentially never used (vowel, frequent consonant, all letters, non-letters) because their leaf nodes were sufficiently frequent to not require fallback. Thus, the original fallback model was essentially a frequency of use model with the infrequent letters falling back to a group average of the infrequent letters.

Two new fallback models were investigated (Ritzmann, in preparation). The first, a touch-type model, was based on the fingers used to strike keys by touch typists (Figures 10 and 11), thinking that this model should be superior to the one described above that is frequency oriented but not particularly relevant to typing. The second was a statistical model that groups keys displaying similar key-strike statistics. The results of the touch-type model were similar to those obtained above but not significantly different. The statistical model was significantly poorer than the other two.

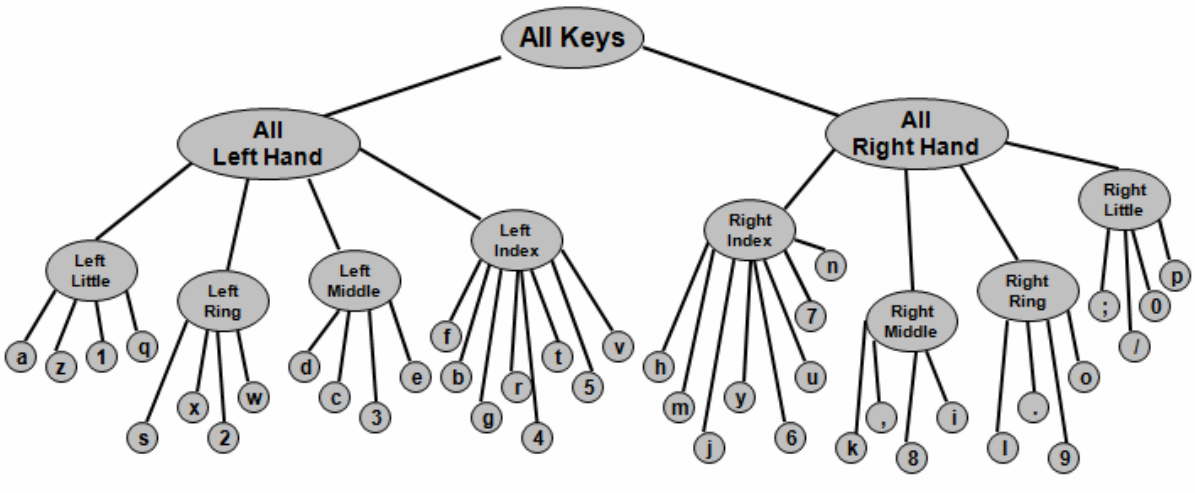


Figure 10. Touch-type hierarchy tree for durations, adopted with permission from Ritzmann (in preparation).

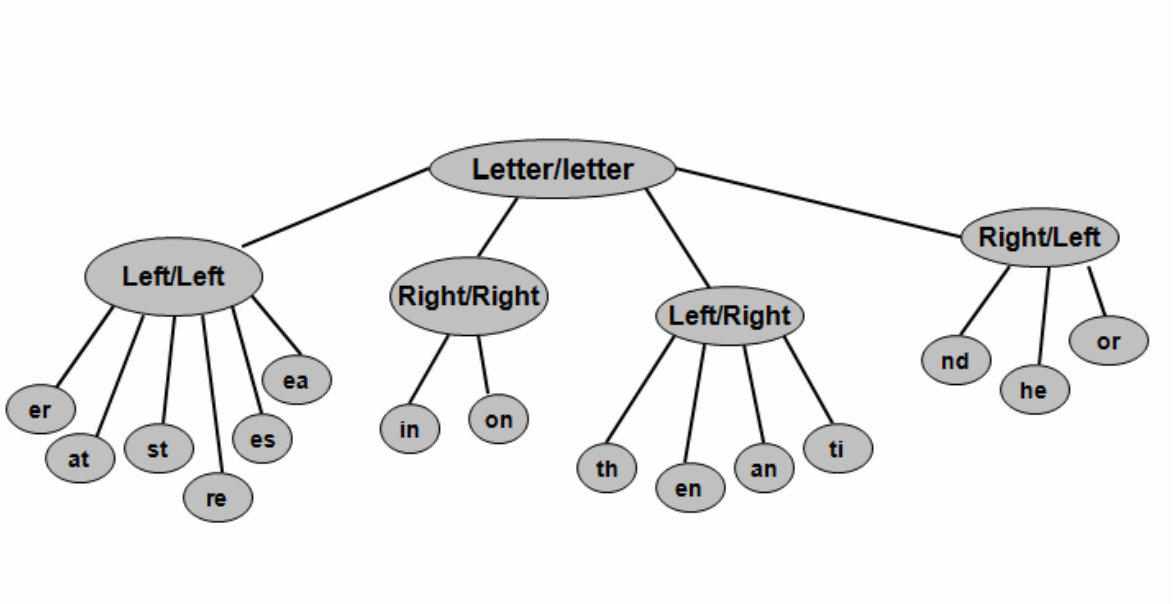


Figure 11. Touch-type hierarchy tree for transitions, adopted with permission from Ritzmann (in preparation).

Outlier parameters. We verified the method of performing outlier removal recursively – that is, continuing to remove outliers until a complete pass through the data resulted in no further outliers being removed (Figure 12). We then measured accuracy as a function of the outlier removal distance (in terms of the number of σ from the μ), finding that the 2σ distance used in the experiments was close to the optimal value of 1.75σ (Figure 13). Note that the parameter settings used in this study were established on different data from an earlier study (Curtin et al., 2006).

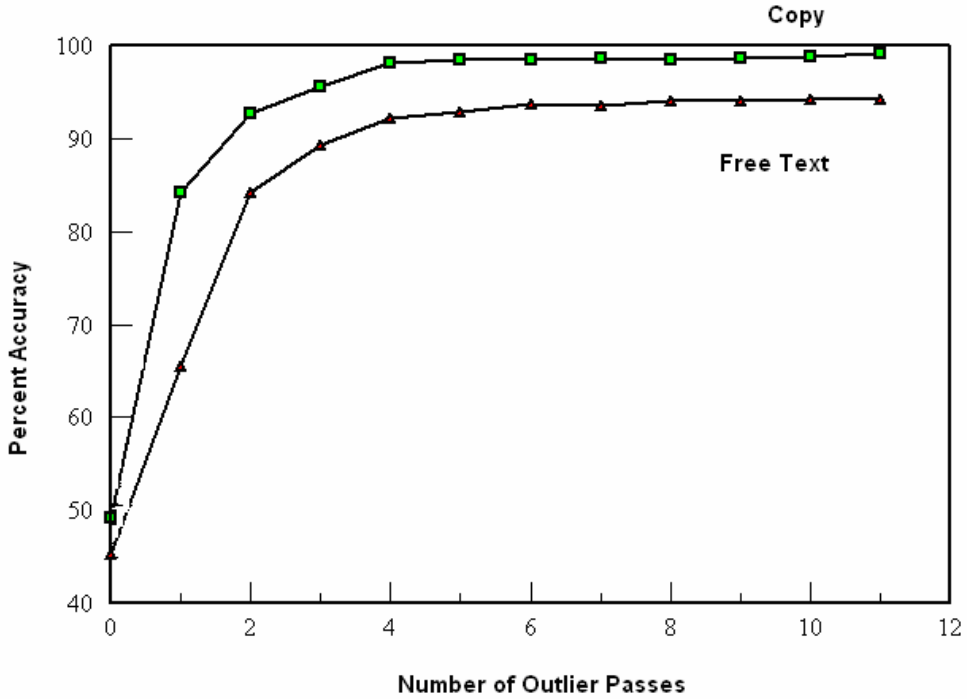


Figure 12. Identification accuracy versus outlier removal passes, adapted with permission from Villani (2006).

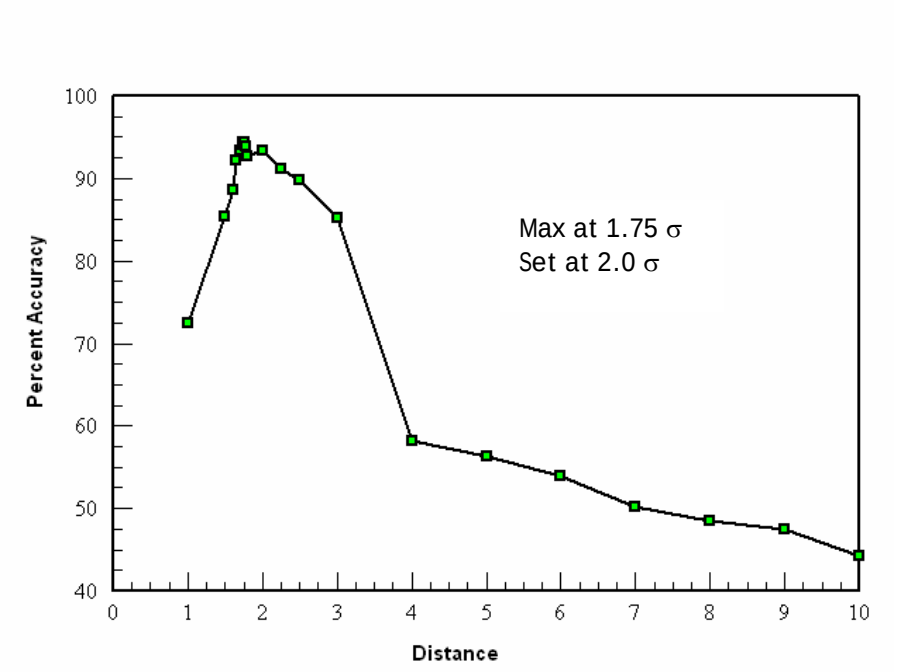


Figure 13. Identification accuracy versus outlier removal distance σ, adapted with permission from Villani (2006).

Number of enrollment samples. In order to check the sufficiency of the number of enrollment samples, we obtained accuracy as the number of enrollment samples varied from one to four (Figure 14). Because each subject supplied five data samples per quadrant, the leave-one-out procedure left a maximum of four enrollment samples to match against. The results indicate that two enrollment samples per user might suffice for this application.

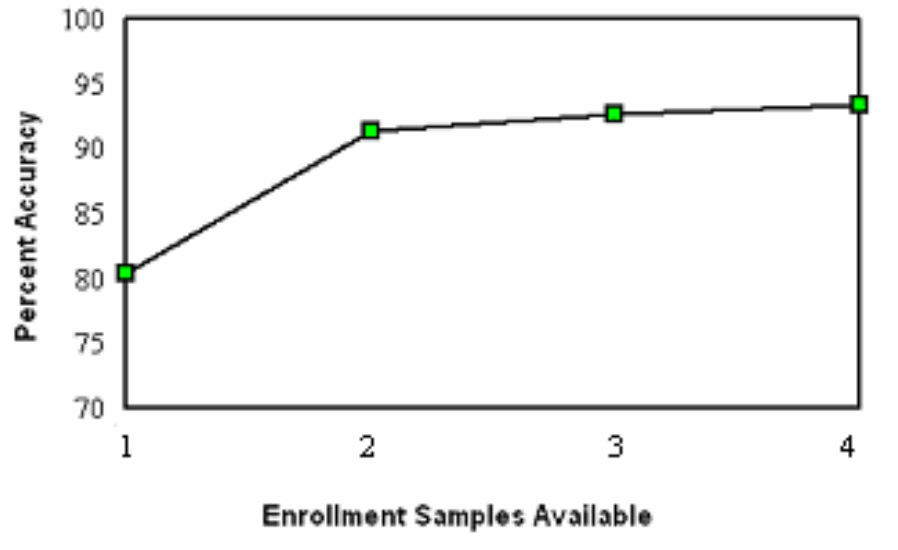


Figure 14. Identification accuracy versus enrollment samples, adapted with permission from Villani (2006).

Input text length. We obtained accuracy as a function of input text length (Figure 15). We found that our choice of 650 keystrokes was in the region where the curve levels off, but that reasonable accuracy can be obtained on shorter text lengths of about 300 keystrokes. The accuracy curve of the copy task is considerably smoother than that of the free text input, perhaps because all copy samples were of the same text but the free text samples were all of different texts.

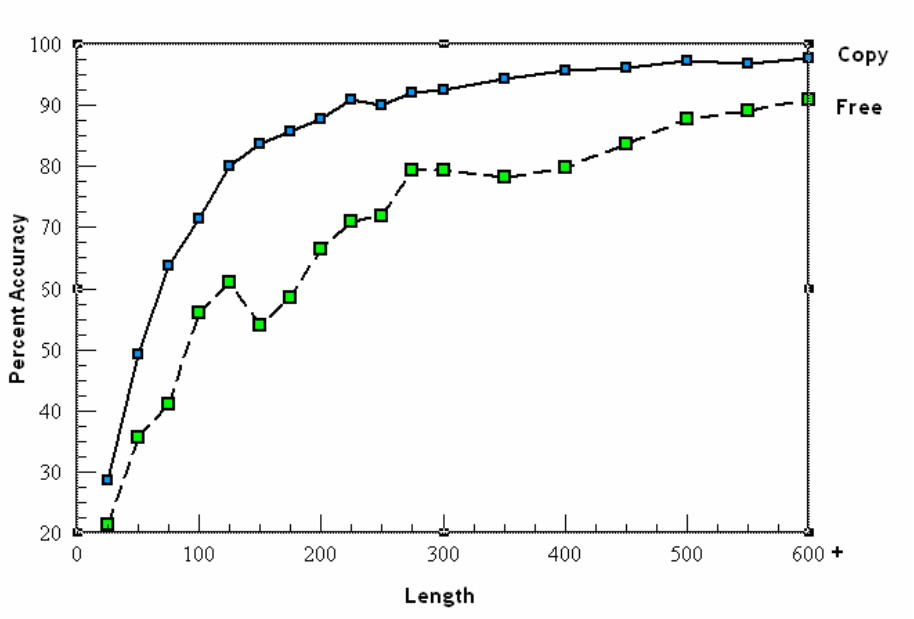


Figure 15. Identification accuracy versus input text length, adapted with permission from Villani (2006).

Probability distributions of statistical features. We verified the normal distribution assumption for the duration and transition times. Figure 16, for example, shows the distributions of the key-press durations for the letter ***u*** for each entry mode.

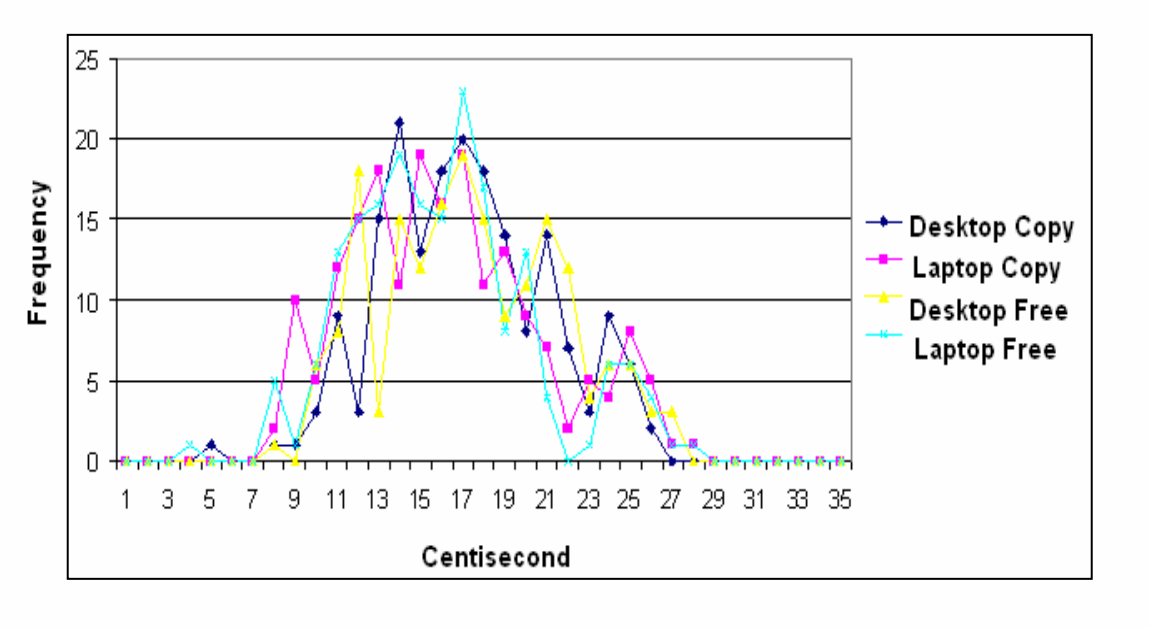


Figure 16. Distributions of "***u***" duration times for each entry mode adapted with permission from Ritzmann (in preparation).

Conclusions and Future Work

The results indicate that the keystroke biometric can be useful for identification and authentication applications if sufficient enrollment samples are available and if the same type of keyboard is used to produce both the enrollment and the questioned samples. The keystroke biometric was significantly weaker for the identification application (but not the authentication application) when enrollment and testing use different input modes (copy or free-text), different keyboard types (desktop or laptop), or both different input modes and different keyboard types. Additional findings include the degree of performance degradation as the number of subjects increases and as the time interval between enrollment and testing increases.

Future work might involve experiments using the system in an actual Internet security situation, like verifying the identity of online test takers. More sophisticated classification techniques such as Support Vector Machines (SVM) might be explored. Also, although it is likely difficult to mimic another person's keystroke pattern, imposter performance might be investigated.

Acknowledgements

We thank the student teams in the masters-level projects course that contributed to this effort over the past several years.

References

Bergadano, F., Gunetti, D., Picardi, C. (2002). User authentication through keystroke dynamics. *ACM Trans. Information & System Security, 5(4),* 367-397.

Bolle, R., Connell, J., Pankanti, S., Ratha, N., & Senior, A. (2004). *Guide to biometrics*. New York: Springer.

Brown, M. & Rogers, S.J. (1993). User identification via keystroke characteristics of typed names using neural networks. *Int. J. Man-Machine Studies, 39(6)*, 999-1014.

Choi, S-S., Yoon, S, Cha, S-H., & Tappert, C.C. (2004). Use of histogram distances in iris authentication, *Proc. MSCE-MLMTA*, Las Vegas, in *Lecture Notes in Computer Science: Image Analysis and Recognition*, New York: Springer, 1118-1124.

Curtin, M., Tappert, C., Villani, M., Ngo, G., Simone, J., St. Fort, H., & Cha, S-H. (2006). Keystroke biometric recognition on long-text input: A feasibility study. *Proc. Int. MultiConf. Engineers & Computer Scientists (IMECS)*, Hong Kong.

Dunn, G. & Everitt, B.S. (2004). *An introduction to mathematical taxonomy.* Dover.

Gaines, H.F., 1956. *Cryptanalysis: A study of ciphers and their solution.* Dover.

Gunetti, D. & Picardi, C. (2005). Keystroke analysis of free text. *ACM Trans. Information & System Security, 8(3)*, 312-347.

Jin, L., Ke, X., Manuel, R., & Wilkerson, M. (2004). Keystroke dynamics: A software based biometric solution. *Proc. 13th USENIX Security Symposium*.

Jurafsky, D. & Martin, J.H. (2000). *Speech and language processing*. New Jersey: Prentice.

Leggett, J. & Williams, G. (1988). Verifying identity via keystroke characteristics. *Int. J. Man-Machine Studies, 28(1)*, 67-76.

Leggett, J., Williams, G., Usnick, M., & Longnecker, M. (1991). Dynamic identity verification via keystroke characteristics. *Int. J. Man Machine Studies, 35(6)*, 859-870.

Montrose, F. & Rubin, A.D. (2000). Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems, 16(4)*, 351-359.

Obaidat, M.S. & Sadoun, B. (1999). Keystroke dynamics based authentication. In *Biometrics: Personal Identification in Networked Society* by Jain, A.K., Bolle, R., & Pankanti, S. (New York: Springer), 213-230.

Peacock, A., Ke, X., & Wilkerson, M. (2004). Typing patterns: A key to user identification. *IEEE Security & Privacy, 2(5)*, 40-47.

Ritzmann, M. (in preparation). Strategies for managing missing or incomplete data in biometric and business applications. Doctoral dissertation, Pace University, New York.

Song, D., Venable, P., & Perrig, A. (1997). User recognition by keystroke latency pattern analysis. Retrieved May, 2005, from http://citeseer.ist.psu.edu/song97user.html.

Villani, M. (2006). Keystroke biometric identification studies on long text input. Doctoral dissertation, Pace University, New York.

Villani, M., Tappert, C., Ngo, G., Simone, J., St. Fort, H., & Cha, H-S. (2006). Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions. *Proc. Computer Vision & Pattern Recognition Workshop on Biometrics*, New York.

Woodward, J.D. Jr., Orlans, N.M., & Higgins, P.T. (2002). *Biometrics*, New York: McGraw-Hill, 107.

Yu, E. & Cho, S. (2004). Keystroke dynamics identity verification – Its problems and practical solutions. *Computers & Security, 23(5)*, 428-440.

Yoon, S., Choi, S-S., Cha, S-H., Lee, Y., & Tappert, C.C. (2005). On the individuality of the iris biometric. *Proc. Int. J. Graphics, Vision & Image Processing, 5(5)*, 63-70.

Appendix: Summary of the 239 features.

Feature	Measure	Feature Measured	Feature	Measure	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	%	sentence 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

List of Figures and Tables