Applying Keystroke Biometrics for User Verification and Identification
Gary Bartolacci, Mary Curtin, Marc Katzenberg, Ngozi Nwana, Sung-Hyuk Cha, and Charles Tappert
Computer Science Department, Pace University
861 Bedford Road, Pleasantville, New York, 10570 USA
garybarto@yahoo.com, {scha, ctappert}@pace.edu
Tel: (914) 773-3989 Fax: (914) 773-3533

Presenter
Gary Bartolacci or Sung-Hyuk Cha

Conference
MLMTA’05

ABSTRACT
Biometrics, the computer-based verification or identification of an individual, is becoming more and more essential due to the increasing demand for high-security systems. Accurate and effective keystroke biometric technology can help provide a major boost to the security of electronic commerce, and it can help curb identify theft. We focus here on two keystroke biometric applications that operate over the Internet and that require a text input of several hundred characters. The first application can help identify a perpetrator of inappropriate or fraudulent Internet activity, and the second can verify the identity of a computer user, such as a student taking an online exam over the Internet. A Java applet collects the raw keystroke data over the Internet. Feature measurements are then extracted from the raw data, and a pattern classification system, initially a simple nearest neighbor classifier, is trained to make the appropriate application-dependent decision. We present preliminary experimental results on the effectiveness of our system in these two applications of keystroke biometrics.

Keywords
Security, keystroke biometric, identity verification, identity recognition

1 INTRODUCTION
The typing biometric, often referred to as keystroke dynamics, examines the way in which a person types or presses keys on a keyboard. It is a behavioral biometric that relies on typing rhythms to verify or recognize the identity of a user on a computer system. Keystroke patterns are currently deployed as a means of hardening passwords in existing computer security schemes. This is due to the fact that an individual’s keystroke patterns tend to be highly repeatable and distinct from other users [3].

The keystroke biometric has been researched to some extent by other investigators. Areas of interest include methods that improve the quality and validity of keystroke analysis. Quality issues include the two categories of keystroke biometric errors (False Accept Rate – FAR, and False Reject Rate - FRR). Gaines, et al. [3] evaluated both of these error rates and Leggett, et al. [13] conducted similar experiments and reported a result of 5.0% FAR and 5.5% FRR on a long string of 537 characters. More recently, Brown and Rogers [11] obtained accuracies of 12%-21% on short name strings, and Obaidat and Sadoun [12] reported a 0% error rate in user verification using 7-character, log-in names. The last two studies used neural networks with both imposter and authentic typing patterns used for training. Also, dynamic shuffling was evaluated as a process to be used on training samples on neural networks as a means of enhancing sample classification and reducing false acceptance or rejection rates during keystroke analysis [11].
The benefits of the keystroke biometric are numerous. They provide a distinct, reproducible, and minimally invasive means of user verification or identification. This is also an inexpensive biometric because the only hardware required is a keyboard. Perhaps most importantly, keystroke dynamics can be used with web-based authentication systems. This is in contrast to physiological biometrics that are often difficult to implement even for internal web-based systems and completely impractical for use in systems accessible on the Internet [2]. Finally, with more businesses moving to e-commerce, the keystroke biometric provides an effective balance between high security and ease of use for customers.

In general, a number of measurements are used in analyzing a user’s typing rhythms. These measurements are usually derived from the raw data of key press times, key release times, and information on which keys are being pressed. From key-press and key-release times, usually measured in milliseconds, a feature vector consisting of keystroke duration times and keystroke transition times can be created [2]. Feature vectors can be collected from all users of a particular system, such as a computer network or web-based system, where keystroke entry is used, and a model that effectively distinguishes an individual from other users of a system can be established.

In this paper, we are concerned with two applications of the keystroke biometric. Both of these applications require significantly more input than a name or password, and we envision these applications operating on text input samples of 200 or more characters. The first is an identification (one-of-n) application. A potential scenario where this application might be applied involves a company in which there has been a problem with the circulation of inappropriate (unprofessional, offensive, or obscene) e-mails from easily accessible desktops in a work environment. If keystroke data can be collected from one of these inappropriate e-mails, a group of potential suspects can then be asked to donate a sample of their typing style. Then, based on a comparison of the text input of these samples to the e-mail in question, an individual might either be eliminated as a suspect or a stronger case made for that individual’s involvement with the offensive document in question.

The second application of interest concerns identity verification, where a binary decision is made as to whether the person is, or is not, the person they claim to be. A potential scenario where this application might be applied is to verify the identity of students taking online quizzes or longer tests. This is important because the student population of online classes is increasing and instructors are becoming more concerned about evaluation security and academic integrity.

In section 2 of this paper, the keystroke biometric system is described. In section 3, the keystroke biometric experiments are described. Finally, section 4 contains the 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. It is important that the user enter the text using his/her normal typing speed and pattern, so that the sample provided is representative of how the user types. If the user is interrupted during data entry, or for any other reason does not enter the text in their usual manner, it is essential that he/she not submit the sample. The “Clear” button will blank all fields, except name and submission number, and allow the user to redo the current entry. Because a vital aspect of this study is the consistency of the samples provided by the subjects, they are asked to key in the text as naturally as possible.

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

- key’s character
- key’s location on the keyboard (1 = standard, 2 = left, 3 = right)
- time the key was pressed (milliseconds)
- time the key was released (milliseconds)
- duration time the key was held down (release time of key [X] – start time of key [X])
- latency time between keys (start time of key [X+1] – the release time of key [X])
- number of left mouse clicks, right mouse clicks, and double clicks during the session

Upon pressing submit, a 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 on Figure 2.

![Figure 1. Java applet for data collection.](image)

![Figure 2: Aligned version of the raw data file for “Hello World!”](image)

### 2.2 Feature Extraction

Feature measurements are computed from the information in the raw data file to obtain a feature vector. A reasonable number of features were chosen to adequately characterize an individual’s keystroke dynamics over writing samples of 200-600 keystrokes. For this study, the feature vector consists of the following 55 measurements:

- The average and standard deviation of the key press durations for the eight most frequent letters of words in the Concise Oxford Dictionary (e, a, r, i, o, t, n, s) [10], together with those of the space cha-
racter and the shift key (20 measurements).

- The average and standard deviations of the transition times between the eight most common letter pairs (digrams) of the alphabet from a portion of the Encyclopedia Britannica (in, th, ti, on, an, he, al, er) [ ], together with those of a space-to-any-letter and any-letter-to-space (20 measurements).
- The total number of key presses for Space, Backspace, Delete, Insert, Home, End, Enter, Ctrl, and all four arrow keys combined (9 measurements).
- The number of key presses for the left and right keyboard locations of the Shift key (2 measurements).
- The total time to enter the text (1 measurement).
- The number of left mouse clicks (1 measurement).
- The number of right mouse clicks (1 measurement).
- The number of double left mouse clicks (1 measurement).

Two preprocessing steps are performed on the features measurements, outlier removal and feature standardization. Outlier removal consists of removing any measurement that is more that two standard deviations from the sample mean. Outlier removal is particularly important in this application because a keyboard user could pause for a phone call, for a sip of coffee, or for numerous other reasons, and the resulting outliers could skew the feature measurements.

After performing outlier removal, we standardize the measurements by converting raw measurement x to x’ by the formula, $x' = (x - x_{\text{min}}) / (x_{\text{max}} - x_{\text{min}})$, where min and max are the minimum and maximum of the measurement over the sample. 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 for the samples in the training set. The subject having the smallest Euclidean distance is identified as the author of the test sample.

3 DESIGN OF EXPERIMENTS

We describe here two experiments. The first is an identification experiment that models, for example, the scenario of the occurrence of an inappropriate e-mail in a company environment. The second is a verification experiment that models online test taking, and is being conducted in an actual course.

For both experiments the applet captures all keystrokes during text entry, including keystrokes used to correct errors. The experiments are also designed so that subjects leave at least a day between entering samples, so that the samples are spread out over time similar to what might be expected in the data taking of an actual application environment.

3.1 Identification Experiment

For the identification experiment, enrollment (training) data entry consists of 10 subjects each keying in a sample text 5 times. The training text consists of approximately 600 characters. For operational (testing) data entry, each of the same 10 subjects keys in the training sample text five more times, and then enters two additional texts, one of similar length and one roughly half the length of the training text, each five times. This structure enables us to compare recognition accuracies on new input of the same text, on input of a different text of the same length, and on input of a different text of shorter length.
3.2 Verification Experiment
The verification experiment is being conducted in a 28-student class of a partially online course in our School of Computer Science and Information Systems at Pace University. The training and testing data consists of various text input samples collected over a period of approximately 10 weeks.

4 RESULTS AND CONCLUSIONS
A reduced version of the identification experiment was conducted using a smaller input text sample, a smaller number of feature measurements, and a smaller number of experimental subjects. On this preliminary version of the experiment we obtained 80% user identification accuracy which we consider promising. We will report complete results with supporting statistics in the final version of the paper.

REFERENCES
1. Lucy Jin et al: Keystroke dynamics: A Software based Biometric Solution: 13th USENIX Security Symposium,
2. S. Cho & E.Yu, j.cose., 2004.02.004). enzhe@snu.ac.kr (E.Yu), zo-on@snu.ac.kr (S. Cho)

6. Peacock, A., Learning User Keystroke Latency Patterns,
8. Fabian Monrose and Aviel D. Rubin. Keystroke dynamics as a biometric for authentication. In Future Generation Computer Systems,
http://avirubin.com/fgcs.pdf
9. IOSoftware. Authentication Basics IOSoftware.
http://www.iosoftware.com/pages/Support/Authentication%20Basics/
Selection%20Process/index.asp
10. AskOxford.com
http://www.askoxford.com/asktheexperts/faq/aboutwords/frequency?view=uk