Proceedings of Student-Faculty Research Day, CSIS, Pace University, May 4th, 2012

Continual Keystroke Biometric Authentication on Short Bursts of Keyboard Input

Ned Bakelman, John V. Monaco, Sung-Hyuk Cha, and Charles C. Tappert
Pace University Seidenberg School of CSIS, White Plains, NY 10606, USA
{nbakelman, vinmonaco}@gmail.com, {scha, ctappert}@pace.edu

This paper was submitted to the IEEE 5th Int. Conf. Biometrics, Washington DC, Oct 2012.

Abstract

This study focuses on intruder detection. Short bursts of keyboard input are analyzed to continually authenticate computer users and verify that they are the authorized ones. The biometric system consists of components for data capture, feature extraction, authentication classification, and ROC curve generation. Experiments were performed on three types of input: text, spreadsheet, and browser. For text input, system performance was obtained as a function of two independent variables: the population size and the number of keystrokes per sample. For each population size, the EER decreased roughly logarithmically as the number of keystrokes per sample was increased. The preliminary results on spreadsheet and browser input indicated considerably weaker performance and further research is required to determine the true biometric value of these types of input.

1. Introduction

This paper describes the development and evaluation of a keystroke biometric system for continual computer-user authentication on short burst-input durations of one or a few minutes. An application of this work is intruder detection, by which we mean the discovery that somebody other than the authentic user is using the computer [8, 9]. Another is verifying the identity of students taking online tests, an application important for the 2008 federal Higher Education Opportunity Act which requires institutions of higher learning to make greater online access control efforts by adopting ubiquitous identification technologies [14]. While intruder detection and online test-taking are similar in terms of authenticating the user, fast discovery is required in the intruder case to prevent significant harm. This study focuses on the intruder detection problem.

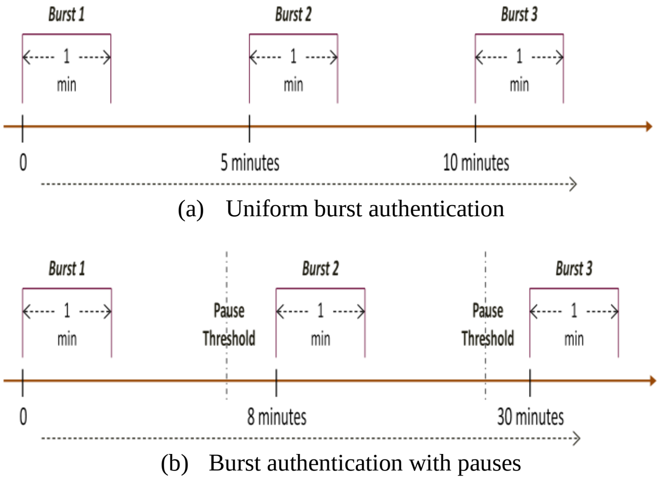
Keystroke biometric systems measure typing characteristics believed to be unique to an individual and difficult to duplicate [6, 16]. The keystroke biometric is one of the less-studied behavioral biometrics and has been reviewed in several articles [17, 28]. The keystroke biometric is appealing for several reasons. First, it is not intrusive, but rather transparent, to computer users who 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 repeated checking after an initial user authentication since keystrokes exist as a mere consequence of using computers [13], and this continuing verification throughout a computer session has been called dynamic verification [21] or active authentication [8].

While most earlier studies used passwords or short name strings [3, 6, 12, 19, 22, 24-26, 28, 29], some used long-text input [4, 13, 21, 23, 27, 30-32, 36]. While most systems developed previously have been experimental in nature, there are a number of commercial keystroke authentication products, primarily for password “hardening” [1, 2, 5, 10, 15, 18].

The remainder of the paper is organized as follows. The next section describes the continual burst authentication strategy. The remaining sections present the methodology, the experimental results, a discussion of the intruder detection problem, and the conclusions.

2. Continual Burst Authentication Strategy

This section defines the terminology and describes the fundamental strategic approach to the problem. *Continual authentication* is ongoing verification but with possible interruptions. This is in contrast to continuous authentication which would mean without interruption. We define *burst authentication* as verification on a short period of computer input *after a pause*. We believe this to be an important concept. One strategy would be to have a moving data interval window that captures, for example, a minute or so of computer input per authentication check occurring at fixed time interval of every five minutes or so (Figure 1 (a)). A better strategy we believe is to only capture the first burst of input after each pause (Figure 1(b)). This is because users often pause for various reasons such as for telephone calls, conversation with colleagues, coffee/bathroom breaks, etc. Furthermore, there would likely be a pause just prior to the entry of an intruder as well. Therefore, only after a pause would re-authentication of the user be required as described in Figure 1 (b).



(a) Uniform burst authentication

(b) Burst authentication with pauses

Figure 1. Burst authentication.

The primary motivation for using this concept of burst authentication is to reduce the frequency of independent authentication checks. This has the advantages of reducing the false alarm rate, avoiding the capture of unnecessarily large quantities of data and using excessive computing resources to process the input, while still providing sufficient data for continual training of the biometric system.

There are two time periods that need to be determined for this strategy. One is the *length of the pause* for burst authentication which needs to be shorter than the entry time of an intruder. Therefore, estimating plausible intruder entry times will provide the critical upper bound on the pause time. Measuring actual authentic user pauses once the system is deployed could additionally provide useful data to determine the potential savings resulting from the reduced authentication frequency of the burst mode relative to the fixed-interval-spacing mode. Note that in an open office environment with computers close together and available to many users, the plausible pause time between an authentic user and an intruder may be negligible, causing the burst authentication approach to revert to the fixed-interval-window-spacing approach.

The second time period of interest is the *length of the data capture authentication window*, which is presumably on the order of a minute or so. For the intruder scenario this needs to be short enough to catch the intruder before significant harm is caused, yet long enough to make an accurate detection and reduce false alarms.

The occurrence of *low-volume computer input* must also be considered. For example, with a user browsing the Internet or checking email while simultaneously engaged in a phone call, the computer input activity may not provide sufficient data for authentication in a short window. Furthermore, in situations, such as phone calls or drinking coffee, in which the user may be using only one hand for keyboard input, the data may be sporadic and not representative of normal user behavior. Fortunately, low-volume computer input of this nature would also not be considered likely intruder behavior. Therefore, data capture windows containing only small quantities of data can probably be safely ignored. The threshold for the quantity of data required for reasonable authentication is therefore an additional parameter to be determined.

Although this study is on the keystroke biometric, the broader plan is to investigate several biometric system components for potential integration into a powerful cyber-security system to provide a multi-level computational behavioral cognitive "fingerprint" of the person operating the computer. For example, keystroke and mouse components operate at the subconscious automatic motor control level, a stylometry component operates at the higher cognitive linguistic (character, word, syntax) level, and an intruder operational behavior component operates at the highest cognitive semantic level of intentional motivation (Figure 2).

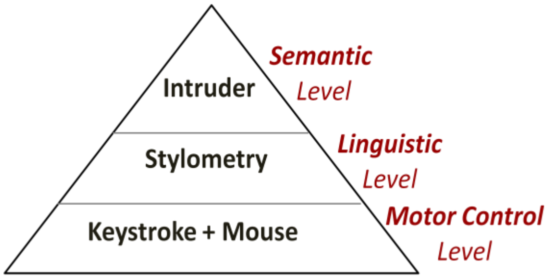


Figure 2. Behavioral biometrics and human cognitive levels.

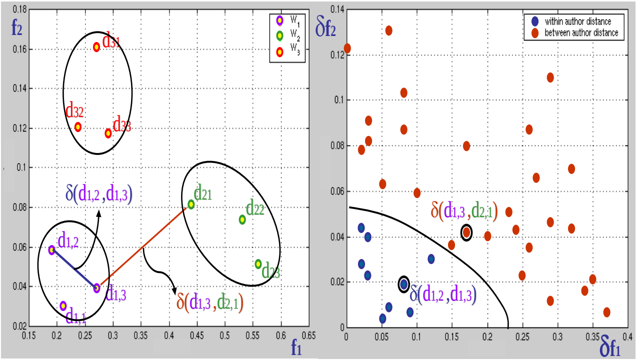
3. Methodology

Given the scenario where an authentic user leaves his system unlocked and unattended, the question becomes how fast and how accurate can the unauthorized use of that computer be detected. Intruder behavior can come in various forms: impersonating the sender of malicious email, modifying financial documents, searching for account codes and passwords to gain access to company secrets, installing malware, etc. These scenarios demonstrate that keyboard input can occur in any of a variety of applications that may be running on the system, and the nature of the input can be different based on context. For instance, input from a user typing email is usually different from input from a user typing into a spreadsheet because the latter data will likely include many numeric keystrokes whereas email may not include any. It is therefore apparent that for keystroke dynamics to be effective at intruder detection the arbitrary nature of the input along with its context need to be factored into the detection scheme.

This work is aimed at developing a behavioral biometric system to continually authenticate users of standard desktop/laptop computers. For the computer environment we target the standard office environment computer that includes keyboard, mouse, Windows operating system, network interface card, connection to a printer, and the standard software product suite of Microsoft Office applications.

This study used an existing system consisting of components for data capture, feature extraction, and authentication classification [30-32]. The keystroke data capture used the PC Windows-event clock, and although the key press and release time were reported in milliseconds, the actual time resolution was 15.6 milliseconds [20]. While the data capturing and feature extraction components have been described in the aforementioned papers, the system backend is important to understand for this study and will be described in some detail.

A vector-difference authentication model transforms a multi-class problem into a two-class problem (Figure 1). The resulting two classes are *within-class* ("you are authenticated") and *between-class* ("you are not authenticated"). This is a strong inferential statistics method found to be particularly effective in large open biometric systems and in multidimensional feature-space problems [7, 35].



(a) Feature space (b) Feature-difference space

Figure 3. Transformation from feature space (a) to feature distance space (b), adapted from [35].

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 3 (a) plots the biometric sample data for these three people in two-dimensional feature space. This feature space is transformed into a feature-difference 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 individual feature vector of the i^{th} person's j^{th} biometric sample, then $x_\oplus$ and $x_\varnothing$ are calculated as follows:

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

where n is the number of people, m is the number of samples per person, and the absolute value is of the elements of these vectors. Figure 3 (b) shows the transformed feature distance space for the example problem.

If n people provide m biometric samples each, the numbers of intra-person and inter-person distance samples, respectively, are [24]:

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

In the authentication process, a user's keystroke sample requiring authentication is first converted into a feature vector. The difference between this feature vector and an earlier-obtained enrollment feature vector from this user is computed, and the resulting difference vector is classified as within-class (intra-person) for authentication or between-class (inter-person) for non-authentication. The *k*-nearest-neighbor method performs this classification by comparing this feature-difference vector against those in the training set.

To obtain system performance we simulate the authentication process of many true users trying to get authenticated and of many imposters trying to get authenticated as other users. This is done by using the numbers of the inter- and intra-person distances explained above. For example, if we have 14 keystroke test samples from each of five users, then (from the equation above) there are 140 intra-person distances to simulate true users and 2275 inter-person distances to simulate imposters. The feature distance space is populated similarly during training.

Receiver operating characteristic (ROC) curves are obtained by using a weighted procedure of the k nearest neighbors [26]. This procedure uses a linear rank weighting, assigning the first choice (nearest neighbor) a weight of k, second a weight of k-1, ... , and the k^{th} a weight of 1. The maximum score when all choices are within-class is $k+(k-1)+...+1 = k(k+1)/2$, and the minimum score is 0. Now, consider that we authenticate a user if the weighted-within-class choices are greater or equal to m, where m varies from 0 to $k(k+1)/2$, and compute the (FRR, FAR) pairs for each m to obtain an ROC curve. The ROC curves in the experimental section below used 21 nearest neighbors to provide weighted scores in the range 0-231 and thus 232 points on the ROC curve.

4. Experimental Results

Three sets of experiments were conducted in this study, each on a different type of keystroke data: text input, spreadsheet input, and browser input.

4.1. Text input experiments

This set of experiments employed free-text (arbitrary input) data samples from the Zack, et al. study [36]. All the data samples contained over 500 keystrokes and were input on Dell desktop PCs and on laptop PCs (almost exclusively Dell machines).

There were two closed system experiments. The first experiment (Train-14/Test-14) had 14 participants with ten samples per user for training and five for testing. The second (Train-30/Test-30) had 30 participants, adding to the first experiment five training and five testing samples from each of 16 additional participants. The data samples for these two experiments were collected in sets of five, the sets recorded at two-week intervals, and the five samples of a set usually recorded in a single day's session. The participants were instructed to enter emails on five different topics (from a given list of topics) for their five samples in a set.

The third experiment (Train-120/Test-30) diluted the training data by adding a single set of five samples from each of 90 additional participants. For these data the participants were simply instructed to enter arbitrary emails. Thus, training was on 120 participants and testing on the same 30-participant data as experiment 2.

For each of these three experiments, a series of sub-experiments was performed by testing on the first 100, 200, 300, 400, and 500 keystrokes of each test sample, and also on the complete test data samples that had an average length of 755 keystrokes. These experiments resulted in three graphs of the EER as a function of the number of keystroke (Figure 4).

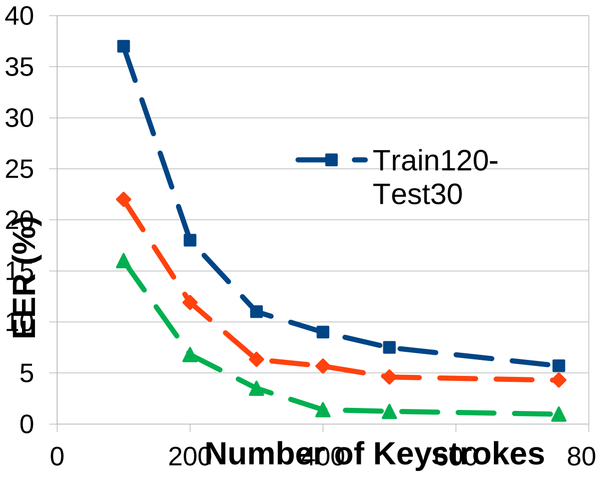


Figure 4. Three experiments: EER versus number of keystrokes.

For the complete data samples that average 755 keystrokes, Figures 5, 6, and 7 show the ROC curves and the FAR/FRR plots for the three experiments. Similar figures were also obtained for the other keystroke lengths (not shown). The crossover points of the FAR and FRR curves as a function of the ROC-curve derivation parameter *m* provided good estimates of the EERs.

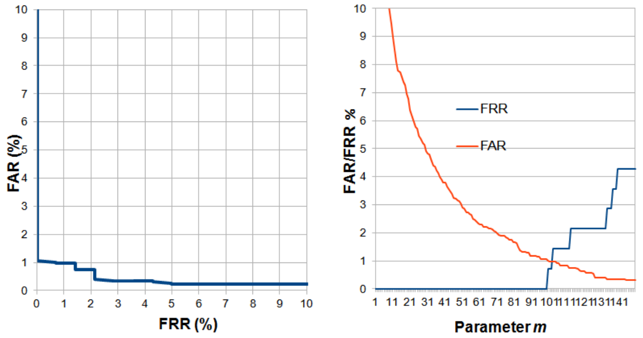


Figure 5. Train-14/Test-14: ROC curve (left), FAR and FRR versus parameter *m* (right).

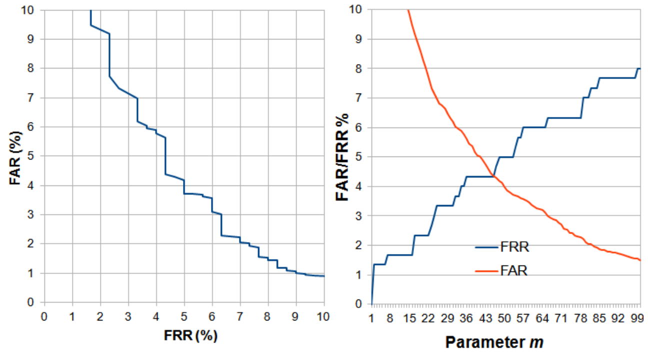


Figure 6. Train-30/Test-30: ROC curve (left), FAR and FRR versus parameter *m* (right).

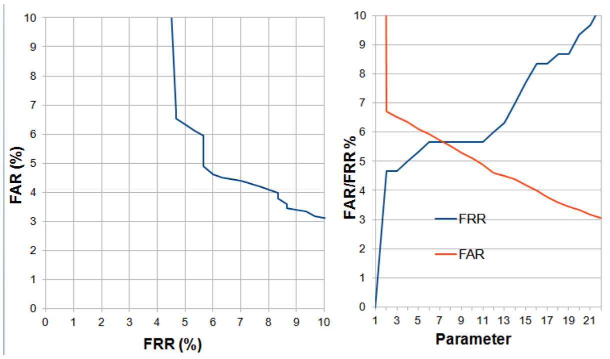


Figure 7. Train-120/Test-30: ROC curve (left), FAR and FRR versus parameter *m* (right).

4.2 Spreadsheet input experiment

For the second and third experiments, input was obtained from a freeware keylogger [11] and the samples were obtained from student participants over a period of several weeks. For the spreadsheet experiment, context specific data samples were captured from a spreadsheet template that simulated a three year comparative balance sheet statement. Although the participants were not given specific input to enter, they were instructed to enter whole numbers, two-decimal-point numbers, and textual journal entries in various areas of the template. Ten samples were collected from each of 20 participants for a total of 200 samples. These samples provided a mix of mostly numeric and some text data. The majority of the participants entered the numeric data from the keypad, while some used the QWERTY portion of the keyboard. There were about 400 keystrokes per sample. A large number of numeric features were added to the system for the spreadsheet experiment, and because their description will not fit into this paper they are available on request from the authors.

Simulating a closed system, for each of the 20 participants five samples were used for training and the other five for testing. The results are shown in Figure 8. The EER was approximately 13.5%.

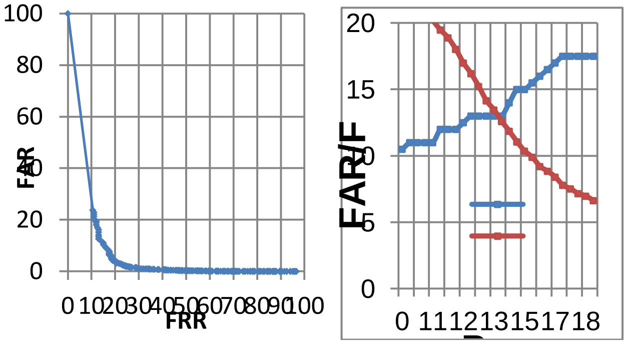


Figure 8. Spreadsheet Train-20/Test-20: ROC curve (left), FAR and FRR versus parameter *m* (right).

4.3 Browser input experiment

For the browser experiment, context specific data samples were captured from Web surfing using a browser. The participants were instructed to use a search engine to lookup stock quotes and simulate online shopping. Ten samples were collected from each of 15 participants. There were fewer than 200 keystrokes per sample with the majority of the input coming from mouse movement data.

Again simulating a closed system, for each of the 15 participants five data samples were used for training and the other five for testing. The results are shown in Figure 9. The EER was approximately 30%.

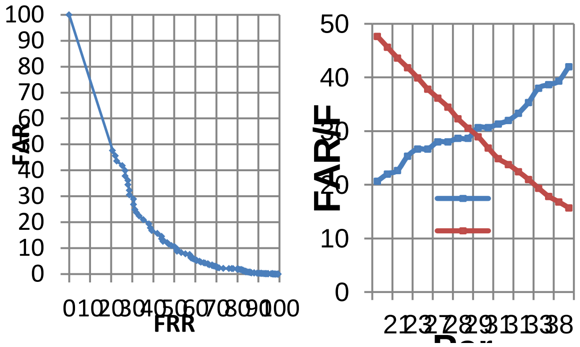


Figure 9. Browser Train-15/Test-15: ROC curve (left), FAR and FRR versus parameter *m* (right).

5. Discussion of Intruder Detection Problem

Considering the intruder detection problem, it is important to relate typing speed to the number of keystrokes per minute. The average word length is five, plus a space, or six characters per word [33]. For average computer users, the average typing speed is 33 words per minute, while a professional typist's speed is about twice that of the average user [34]. Since the number of keystrokes is usually only slightly more than the number of characters, the average computer user generates about 200 keystrokes per minute, while a professional typist, and presumably a fast typing intruder, about 400 keystrokes per minute. Therefore, a single minute of a potential intruder's burst input would likely be in the 200-400 keystroke range, which is centered in the range covered in the text input experiments (Figure 4). With the knees of the curves at roughly 300 keystrokes, a small difference in the length of the data capture window, say from 1.0 to 1.5 minutes, could make a rather large difference in the performance.

6. Conclusions

The main contribution of this study was the evaluation of the text-input performance as a function of two independent variables – the population size and the number of keystrokes per test sample. As the number of keystrokes per sample was increased the EER decreased roughly logarithmically.

Another finding was that the text input data seemed to provide more robust biometric information than the spreadsheet and browser input data. For example, the spreadsheet input experiment with 20 participants and 400 keystrokes per sample yielded an EER of 13.5%, whereas a corresponding text input experiment might be expected to yield an EER of about 4% according to the graphs in Figure 4. Similarly, the browser input experiment with 15 participants and somewhat less than 200 keystrokes yielded an EER of 30%, whereas a corresponding text input experiment might be expected to yield an EER of about 10% according to the graphs in Figure 4. This lower performance on the spreadsheet and browser data relative to that on the text data might be explained by considering that text input is a more highly practiced skill and therefore tends to be more stable from sample to sample. Regarding the spreadsheet experiment, perhaps the participants lacked sufficient skill in working with spreadsheets or found it difficult to work with the spreadsheet template or needed sufficient practice in working with the template. Regarding the browser experiment, because most of the data obtained was related to mouse activity, the small amounts of text input may not have been input in a normal typing text-input manner, and possibly even entered with one hand. In any case, these experimental results on the spreadsheet and browser input are very preliminary and further work is required to determine the biometric value of these types of input.

To obtain system performance in this study we simulated the authentication process of many true users trying to get authenticated and of many imposters trying to get authenticated as other users. An important advantage of this vector-difference model is that it provides relatively large numbers of inter- and intra-person distance samples for analysis and ROC curve generation. However, the drawback to using all the possible vector-difference pairs in this manner is that the simulated authentication decision is based on only one vector difference of the feature vector to be authenticated against a feature vector of the same person for authentication or against a different person for non-authentication to check for imposters. In an actual authentication system, however, the feature vector to be authenticated should be matched against several feature vectors (templates) of the authentic user in making the authentication decision, and this will be explored in future experiments.

In this study the EER was used for simplicity as a single value of performance to show the trends of performance as a function of the population size and the number of keystrokes per sample. However, in a deployed system the operating point on the ROC curve would be chosen appropriately, usually with a considerably lower FAR than FRR. Although a low FAR operating point would incur more false rejections, repeated authentication failures could be required before deciding on a strong non-authentication.

Future work on intruder detection should also focus directly on the type of input expected from intruders, such as specific commands entered from a command prompt. These might include DOS commands (cd, dir, copy, del, systeminfo, regedit, etc.), UNIX commands (ls, cp, rm, whoami, chmod, ipconfig, etc.), and executable file extensions (exe, com, dll, etc.). Because an intruder will likely interact with the GUI, the biometric value of mouse information – context, clicks, trajectory, speed, and acceleration – should also be explored.

This study investigated the detection of intruders on standard PCs. It is anticipated that this work, when fully extended to cover all the aforementioned interrelated biometrics, should have the capability of detecting unauthorized users of computers in different environments such as government offices and private sector workplaces. Accordingly, an effective real time keystroke verification system that can authenticate early and often can determine for instance if an individual swapped places during the taking of an online exam or detect whether an unauthorized user is suddenly working on a machine he/she is not supposed to. In situations that deal with sensitive materials such as the military or the government, this can be of vital importance. Similarly, systems that deal with personal health information (in hospitals, nursing homes, etc.) can provide an additional layer of security with this approach thereby averting or reducing the risk of an unwanted user entering or requesting sensitive health related data.

As the Internet continues to grow in size and in use, measures that can successfully authenticate on a continual basis, and verify that you are who you say you are, can be of vital importance in the years ahead.

Acknowledgements

The authors thank the student teams that were involved in collecting the data and developing various parts of the systems involved in these experiments.

References

[1] AdmitOneSecurity. (Apr 2012). http://www.admitonesecurity.com/

[2] AuthenWare. (Apr 2012). http://www.authenware.com/

[3] S. Bender and H. Postley, "Key sequence rhythm recognition system and method," US Patent 7,206,938, 2007.

[4] F. Bergadano, D. Gunetti, and C. Picardi, "User Authentication through Keystroke Dynamics," *ACM Trans. Inf. Syst*, 2002, pp. 367-397.

[5] bioChec. (Apr 2012). http://www.biochec.com/

[6] R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior, *Guide to biometrics*. New York: Springer, 2004.

[7] S. Cha and S. Srihari, "Writer Identification: Statistical Analysis and Dichotomizer," in *Advances in Pattern Recognition*. vol. 1876: Springer, 2000, pp. 123-132.

[8] DARPA. (Apr 2012). *Active Authentication Program*. https://www.fbo.gov/index?s=opportunity&mode=form&id=c7968647352f0276fc1b28817c581d86&tab=core&_cview=0

[9] DARPA. (April, 2012). *Cyber Genome Program*. https://www.fbo.gov/index?s=opportunity&mode=form&id=c34caee99a41eb14d4ca81949d4f2fde&tab=core&_cview=0

[10] DeepnetSecurity. (Apr 2012). http://www.deepnetsecurity.com/

[11] E. Fimbel. (July 29). *keyloggerbasiclabbook - basiclabbook*. http://sites.google.com/site/basiclabbook/keyloggerbasiclabbook

[12] R. Giot, M. El-Abed, and C. Rosenberger, "Keystroke dynamics with low constraints svm based passphrase enrollment," IEEE Int. Conf. Biometrics (BTAS 2009), 2009.

[13] D. Gunetti and C. Picardi, "Keystroke Analysis of Free Text," *ACM Transactions on Information Systems*, vol. 8, 2005, pp. 312-347.

[14] HEOA. (May, 2011). *Higher Education Opportunity Act (HEOA) of 2008*. http://www2.ed.gov/policy/highered/leg/hea08/index.html

[15] IDControl. (Apr 2012). http://www.idcontrol.com/

[16] L. Jin, X. Ke, R. Manual, and M. Wilkerson, "Keystroke dynamics: A software based biometric solution," *13th USENIX Security Symposium*, 2004.

[17] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing J.*, vol. 11, 2011.

[18] KeyTrac. (Apr 2012). http://www.keytrac.de/

[19] K. Killourhy and R. Maxion, "Comparing Anomaly-Detection Algorithms for Keystroke Dynamics," *Int. Conf. Dependable Systems & Networks (DSN-09)*, Lisbon, Portugal, 2009, pp. 125-134.

[20] K. Killourhy and R. Maxion, "The Effect of Clock Resolution on Keystroke Dynamics," in *Raid 2008, LNCS*. vol. 5230, R. Lippmann, E. Kirda, and A. Trachtenberg, Eds.: Springer, 2008, pp. 331-350.

[21] F. Leggett, G. Williams, and M. Usnick, "Dynamic Identity Verification Keystroke Characteristics," *Int. J. Man-Machine Studies*, 1991, pp. 859-870.

[22] Y. Li, B. Zhang, Y. Cao, S. Zhao, Y. Gao, and J. Liu, "Study on the BeiHang Keystroke Dynamics Database," Int. Joint Conf. Biometrics (IJCB 2011), Washington, D.C., 2011.

[23] A. Messerman, C. Mustafi, S. Camtepe, and S.Albayrak, "Continuous and Non-intrusive Identity Verification in Real-time Environments based on Free-Text Keystroke Dynamics," *Int. Joint Conf. Biometrics (IJCB 2011)*, Washington D.C., 2011.

[24] F. Monrose, M. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," *Int. J. Information Security*, vol. 1, 2002, pp. 69-83.

[25] F. Monrose and A. Rubin, "Keystroke Dynamics as a Biometric for Authentication," *Future Generation Computer Systems*, vol. 16, 2000, pp. 351-359.

[26] M. Obaidat and B.Sadoun, "Keystroke dynamics based authentication," in *Biometrics: Personal Identification in Networked Society by Jain, et. al.*, New York: Springer, 1999, pp. 213-230.

[27] A. Peacock, X. Ke, and M. Wilkerson, "Typing patterns: A key to user identification," *IEEE Security & Privacy*, vol. 2, 2004, pp. 40-47.

[28] K. Revett, "Chapter 4: Keystroke dynamics," in *Behavioral biometrics: A remote access approach*: Wiley, 2008, pp. 73-136.

[29] R. Rodrigues, G. Yared, C. Costa, J. Yabu-Uti, F. Violaro, and L. Ling, "Biometric access control through numerical keyboards based on keystroke dynamics," in *Lecture notes in computer science*. vol. 3832/2005, 2005, pp. 640-646.

[30] C. Tappert, S. Cha, M. Villani, and R. Zack, "A Keystroke Biometric System for Long-Text Input," *Int. J. Info. Security and Privacy*, 2010, pp. 32-60.

[31] C. Tappert, M. Villani, and S. Cha, *Keystroke Biometric Identification and Authentication on Long-Text Input*: IGI Global, 2009.

[32] M. Villani, C. Tappert, G. Ngo, J. Simone, H. S. Fort, and S. Cha, "Keystroke Biometric Recognition Studies on Long-Text Input Under Ideal and Application-Oriented Conditions," *Computer Vision & Pattern Recognition Workshop on Biometrics*, New York, 2006.

[33] Wikipedia. (Apr 2012). *Size comparisons*. http://en.wikipedia.org/wiki/Wikipedia:Size_comparisons

[34] Wikipedia. (Apr 2012). *Words per minute*. http://en.wikipedia.org/wiki/Words_per_minute

[35] S. Yoon, S.Choi, S. Cha, Y. Lee, and C. Tappert, "On the Individuality of the Iris Biometric," *Int. J. Graphics, Vision and Image Processing*, 2005, pp. 63-70.

[36] R. Zack, C. Tappert, and S.Cha, "Performance of a Long-Text-Input Keystroke Biometric Authentication System Using an Improved k-Nearest-Neighbor Classification Method," *IEEE 4th Int Conf Biometrics (BTAS 2010)*, Washington D.C., 2010.