Developing a Keystroke Biometric System for Continual Authentication of Computer Users

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

***Abstract*—Data windows of keyboard input are analyzed to continually authenticate computer users and verify that they are the authorized ones. Because the focus is on fast intruder detection, the authentication process operates on short bursts of roughly a minute of keystroke input, while the training process can be extensive and use hours of input. The biometric system consists of components for data capture, feature extraction, authentication classification, and receiver-operating-characteristic curve generation. Using keystroke data from 120 users, system performance was obtained as a function of two independent variables: the user population size and the number of keystrokes per sample. For each population size, the performance increased (and the equal error rate decreased) roughly logarithmically as the number of keystrokes per sample was increased. The best closed-system performance results of 99 percent on 14 participants and 96 percent on 30 participants indicate the potential of this approach.**

Keywords-pattern recognition, machine learning, biometrics, keystroke biometrics, intruder detection, user authentication

I. 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, usually relegated to conference sessions on "other biometrics" and described only briefly in books on biometrics. Nevertheless, the keystroke biometric has been reviewed in several recent 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].

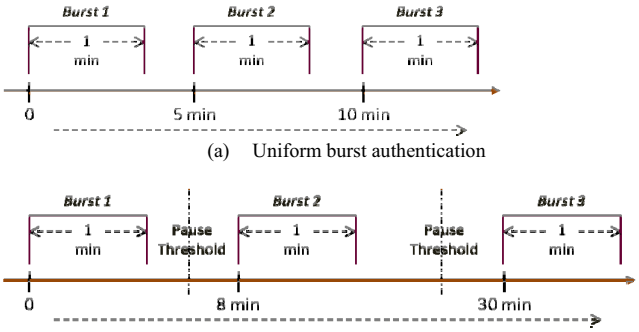
While most earlier studies used passwords or short name strings [3, 6, 12, 19, 22, 24-26, 28, 29], some used long-text input [4, 13, 21, 23, 27, 30-32, 36]. While most systems developed previously have been experimental in nature, there are a number of commercial keystroke authentication products, primarily 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, and the conclusions.

II. CONTINUAL BUSRT 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 (Fig. 1 (a)). A better strategy we believe is to only capture the first burst of input after each pause (Fig. 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 Fig. 1 (b).

978-0-7695-4782-4/12 $26.00 © 2012 IEEE
DOI 10.1109/EISIC.2012.58



(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.

For this application of rapid intruder detection, the authentication process operates ideally on short bursts of a minute or so duration of text input. However, because huge quantities of authentic user data are usually available, the training process can be extensive and use many hours of input.

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 (Fig. 2).

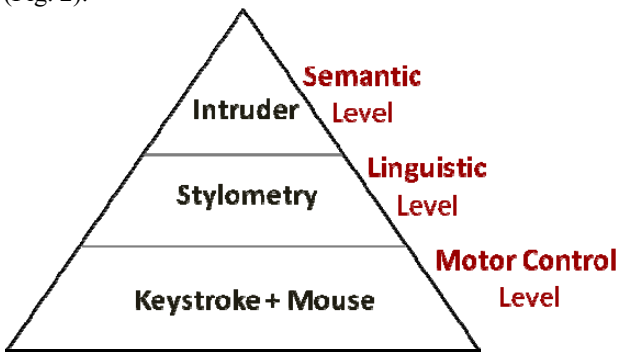


Figure 2. Behavioral biometrics and human cognitive levels.

For using such a continual authentication system on government or private company machines, keylogger software could be installed to transparently capture user input on all monitored PCs and the authentication processing performed on servers. However, because many employees like to use their PC for occasional personal use – email, banking, stock market transactions, etc. – there are obvious privacy concerns with a keylogger capturing all input, including account numbers and passwords. And, although the organizations might say they can monitor their machines as they like, the employees could have strong objections. To increase user acceptance and ameliorate privacy concerns, monitored machines should be clearly marked as such and unmonitored machines could be made available for employee personal use during lunch and break times. Nevertheless, privacy concerns remain.

III. METHODOLOGY

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 were captured in a Java applet that used the PC Windows-event clock. Although the key press and release times were recorded in a millisecond format, the actual time resolution of the PC Windows-event clock was recently discovered to be only 15.6 milliseconds [20].

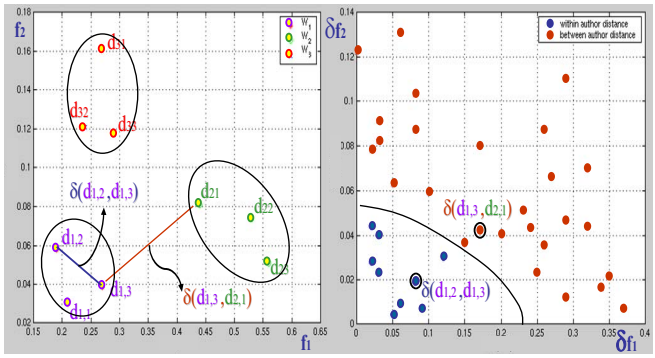
The feature extraction component extracts a vector of 239 features from the raw timing data. The features are statistical in nature and designed to characterize an individual's keystroke dynamics over writing samples of 200 or more characters. Most of the features are averages and standard deviations of key press duration times and of digraph transition times. While key press duration and transition times are typically used as features in keystroke biometric password authentication systems, 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 in email. The features are grouped as follows (see [30] for details):

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

The computation of a keystroke-duration mean or standard deviation requires special handling when there are few samples. For example, when the number of samples for a keyboard key is less than a threshold, the mean is calculated as the weighted average of the mean of the key in question and the mean of the appropriate fallback group of keys at the next highest node in a hierarchy tree. Because we are dealing with long-text input, fallback is necessary for only infrequently used keys. Thus, we ensure 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 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 – overly long transition times – would skew the feature measurements. Overly long key presses can also occur but are rare. Outlier removal consists of removing any duration or transition time that is more than two standard deviations from the mean values. After outlier removal, averages and standard deviations are recalculated recursively until no further outliers can be removed. After performing outlier removal, the feature measurements are standardized into the range 0-1 by clamping each measurement at plus and minus two standard deviations over all samples from all participants. This standardization method gives each measurement roughly equal weight. The feature measurements, the hierarchical trees, the fallback procedure, and the preprocessing steps have been described more fully in earlier papers [30, 31].

The system backend is important to understand for this study and will be described in detail. A vector-difference authentication model transforms a multi-class problem into a two-class problem (Fig. 3). The resulting two classes are *within-person* ("you are authenticated") and *between-person* ("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. Fig. 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 (*within-person distances*, denoted by $x_{\oplus}$) and distances between pairs of samples of *different* people (*between-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:

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

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. Fig. 3 (b) shows the transformed feature distance space for the example problem.

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

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

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-person for authentication or between-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 between- and within-person distances explained above. For example, if we have five keystroke test samples from each of 30 users as in one of the experiments below, then (from the equation above) there are 300 within-person distances to simulate true users and 10,875 between-person distances to simulate imposters. The feature distance space is populated similarly during training.

Receiver operating characteristic (ROC) curves are usually used to characterize the performance of a biometric system, and they show the trade-off between the False Accept Rate (FAR) and the False Reject Rate (FRR). In this study, the 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-person 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-person 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 Equal Error Rate (EER), a common single measure of system performance, is where the FAR equals the FRR. 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.

IV. Experimental Results

This study employed free-text (arbitrary input) keystroke data samples from the Zack, et al. study [36]. Data samples were available from 120 experimental participants: fifteen data samples from each of 14 participants, ten from each of 16 participants, and five from each of 90 participants.

All the data samples contained over 500 keystrokes and averaged 755. They were input on Dell desktop PCs and on laptop PCs (almost exclusively Dell machines).

Four new experiments were conducted on these data – two closed-system and two open-system experiments. The first experiment (Train-14/Test-14) used the 14 participant data: ten samples per user for training and five for testing. The second experiment (Train-30/Test-30) used data from 30 participants, adding to the first experiment five training and five testing samples from each of the 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. These two experiments are closed-system experiments because all the participants are contained in both the training and the test sets. Nevertheless, the system was still tested for intruders because data samples from each participant were matched against samples from the other participants to simulate potential intruders.

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. These 90 participants were simply instructed to enter arbitrary emails. Thus, training was on 120 participants and testing was on the same 30-participant data as experiment 2.

The fourth experiment (Train-75/Test-75) split the 90 participant data into two parts, adding the data from 45 participants to the training set and the other 45 participants to the test set of the second experiment. For validation, a second variation of this experiment swapped the two 45 participant portions of the data, and the results from the two variations were averaged. These experiments perhaps provided a more realistic evaluation of the system because the samples from 45 of the test users were not included in the training set and were never seen before by the system.

For each of these experiments, training used the full data samples of roughly 755 keystrokes per sample. Testing used an independent set of data and each experiment consisted of a series of sub-experiments on the first 100, 200, 300, 400, and 500 keystrokes of each test sample, and also on the full test data samples.

The primary results of these four experiments are shown in the four graphs of the EER as a function of the number of keystrokes (Fig. 4). In this study, the EER was obtained from the ROC curve or more accurately from the FAR and FRR curves versus the parameter *m* (see below). Because system performance (accuracy) and EER sum to one (or 100%), system performance equals 1 - EER.

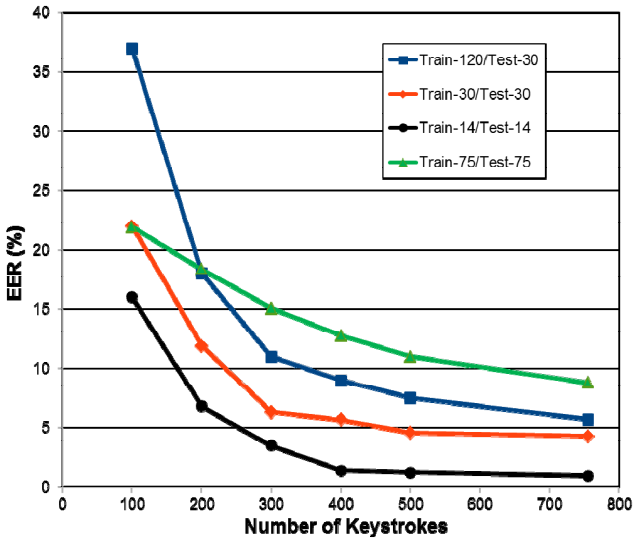


Figure 4. EER versus number of keystrokes.

Figs. 5 and 6 show the ROC curves and the FAR/FRR plots for the last sub-experiment of each of the two closed-system experiments. Similar figures were also obtained for the other keystroke lengths and other experiments (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. Note that because there are many more between-person distances simulating imposters, the FAR curves are considerably smoother than the FRR curves.

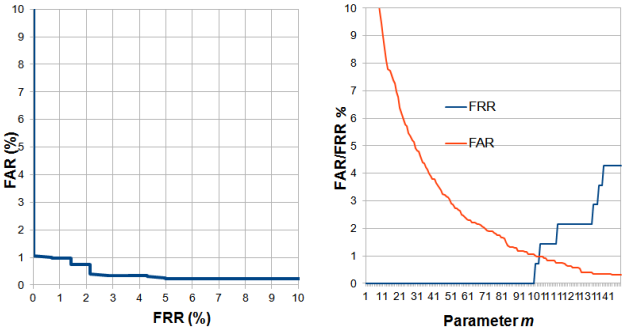


Figure 5. Train-14/Test-14: ROC curve (left), FAR and FRR versus parameter *m* (right), EER = 1%.

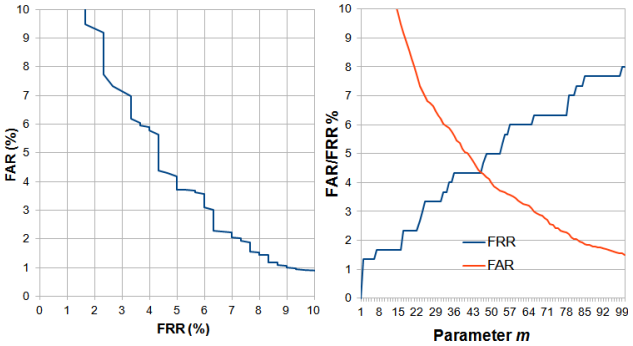


Figure 6. Train-30/Test-30: ROC curve (left), FAR and FRR versus parameter *m* (right), EER = 4%.

V. 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 – after training on the longest keystroke samples available. As the number of keystrokes per test sample was increased, the EER decreased roughly logarithmically, and the EER increased with the increase in population size. The performance results of 99% on 14 participants and 96% on 30 participants indicated the strong potential of this approach.

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 generates about 400 keystrokes per minute. Although one tends to think of an intruder as a fast typist, a safer assumption is that he would be in the average to fast-typist range. 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 (Fig. 4). Because the knees of the curves are at roughly 300 keystrokes and 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, a 1.5 minute or more data-capture window seems appropriate.

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 between- and within-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, several authentication failures could be required before signaling an intruder alarm.

While large quantities of keystroke training data can be collected from the authentic users (over many days, weeks, and even months), the quantity of keystroke data available for detecting unauthorized users must be limited to a minute or so in order to detect the intruder before significant harm is committed. Because large quantities of training data are usually available for this application, elaborate and possibly sophisticated procedures for training the system on significant quantities of data should be investigated.

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 use. In situations that deal with sensitive military or government information, 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.

ACKNOWLEDGMENT

The authors thank the various masters-level project teams that gathered the data samples and coded some of the early versions of the algorithms.

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. (2010). *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 2010). *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 Trans. 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, 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," Chapter 16 (pp. 342-367) in *Behavioral Biometrics for Human Identification*, Edited by Wang and Geng, IGI Global, 2010.
[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.