Correspondence

User Authentication Through Typing Biometrics Features

Lívia C. F. Araújo, Luiz H. R. Sucupira Jr., Miguel G. Lizárraga, Lee L. Ling, and João B. T. Yabu-Uti

***Abstract*—This paper uses a static keystroke dynamics in user authentication. The inputs are the key down and up times and the key ASCII codes captured while the user is typing a string. Four features (key code, two keystroke latencies, and key duration) were analyzed and seven experiments were performed combining these features. The results of the experiments were evaluated with three types of user: the legitimate, the impostor and the observer impostor users. The best results were achieved utilizing all features, obtaining a false rejection rate of 1.45% and a false acceptance rate of 1.89%. This approach can be used to improve the usual login-password authentication when the password is no more a secret. This paper innovates using four features to authenticate users.**

***Index Terms*—Biometrics, keystroke dynamics, pattern recognition, typing biometrics.**

I. Introduction

The login-password authentication is the most usual mechanism used to grant access because it is low-cost, besides its familiarity to a lot of users. However, this authentication is fragile when there is a careless user and/or a weak password. The purpose of this paper is to improve the login-password authentication using biometric characteristics. Biometric characteristics are unique to each person and have advantages as they could not be stolen, lost, or forgotten [1].

The biometric technology employed in this paper is the typing biometrics, also known as keystroke dynamics. Typing biometrics is a process that analyzes the way a user types at a terminal by monitoring the keyboard inputs in attempt to identify users based on their habitual typing rhythm patterns. The typing biometrics authentication can be classified as static or continuous. The static approach analyzes inputs just in a particular moment, and the continuous one analyzes inputs during all user's session [2].

The methodology of this paper is low cost (using a conventional keyboard) and unintrusive (using a password or a login) and uses a static approach (using the login session).

This paper is organized as follows. In Section II, the related studies published are detailed. In Section III, the methodology is explained. In Section IV, the experiments are presented and discussed. Finally, Section V presents the conclusions and future works.

II. Related Work

Some research has been published [2]–[14] in the authentication via typing biometrics since 1990. Some aspects presented in these works are detailed as follows.

- *Target String*: It is the input string that will be typed by the user and monitored by the system. In [3], four strings (login, password, first name, and last name) were used as targets. In some works, the password itself was employed. String length is a very important issue, considering that in [4] it was stated that misclassification increases as the string length drops to fewer than ten characters.
- *Number of Samples*: Samples are collected during the enrollment process to compound the training set. Its number varies a lot, since it was as few as three samples per user in [5] and as many as 30 samples per user in [6]. In [2], it was concluded that fewer than six samples is not recommended to obtain good performance.
- *Features*: Two of the most used features are duration of the key, that is the time interval that a key remains pressed [5], and keystroke latency, that is the time interval between successive keystrokes [5]. In [7]–[9], the combination of these features resulted in better results than using them in isolation. De Ru *et al.* [10] analyzed a feature based on the distance of the keys in the keyboard and the combination of keys called typing difficulty, besides the keystroke latency;
- *Timing Accuracy*: As most of the typing biometrics features are time-based, the precision of the key-up and the key-down times have to be analyzed. The timing accuracy varies between 0.1 ms [7] and 1000 ms (1 s) [11].
- *Trials of Authentication*: In [12], it was observed that legitimate users usually fail in the first trial of authentication, but in the second one, a successful authentication was realized. In [6], each user must type his target string two times using a shuffling technique.
- *Adaptation Mechanism*: Biometric characteristics change over time. An adaptation mechanism or a re-enrollment could be performed to maintain the templates updated. Most of the researchers did not mention this issue, but, in [13], an adaptation mechanism was used. This mechanism creates a new updated template every time a successful authentication is performed, including the new sample and discarding the oldest one.
- *Classifier*: In [2]–[4], [6], [7], and [11], a statistical classifier was applied, using known techniques as k-means, Bayes, etc. In [9] and [10], fuzzy logic was applied using a user's categorization as output. Finally, in [5], [8], and [14], neural networks were experimented with, although in [2], it was explained that this classifier is not appropriate to access control systems because of training requirements (e.g., time consuming). In [12], a statistical, a neural network, and a fuzzy classifier were combined.

A. Our Approach

A target string with at least ten characters will be used according to [4]. In the enrollment, ten samples were collected from each user. We observe that more than ten samples in enrollment annoy the users. The features analyzed are the key code, two keystrokes latencies, and the key duration. The combination of these four features is novel in the typing biometrics research area. A 1-ms timing accuracy is used compatible with the collected data (the details are presented in the methodology section). An adaptation mechanism is used to maintain the updated template. A statistical classifier is applied since neural networks are not appropriate to this approach and a fuzzy classifier was already analyzed using our methodology in [9].

Manuscript received November 1, 2003; revised September 28, 2004. This work was supported in part by FAPESP, CNPq, and CAPES. The associate editor coordinating the review of this paper and approving it for publication was Dr. Anil K. Jain.

The authors are with the School of Electrical and Computer Engineering, State University of Campinas, Campinas, Brazil (e-mail: liviacris@hotmail.com; luigijr@yahoo.com; lizarrag@decom.fee.unicamp.br; lee@decom.fee.unicamp.br; yabuuti@decom.fee.unicamp.br).

Digital Object Identifier 10.1109/TSP.2004.839903

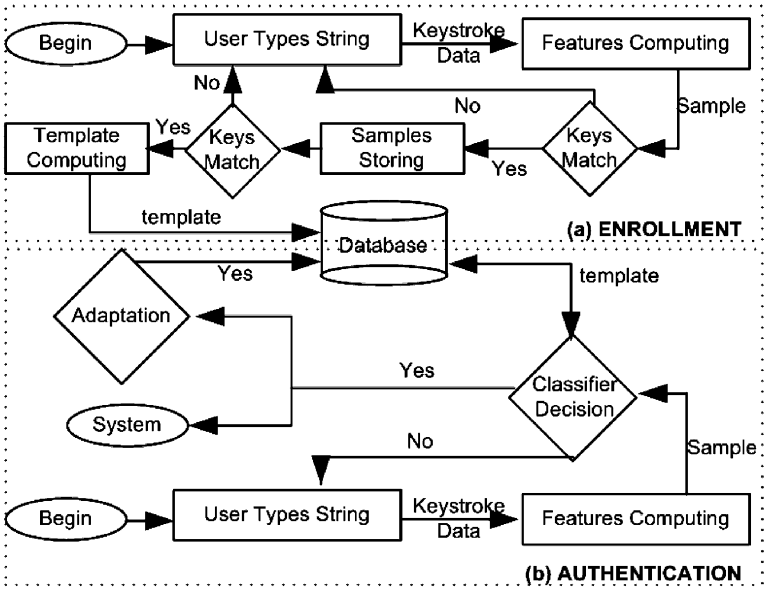

Fig. 1. Flowchart of the Methodology. (a) Enrollment. (b) Authentication.

III. Methodology

The flowchart of the main steps of the methodology can be visualized in Fig. 1. This methodology does not deal with typographic errors, and a session of authentication begins when there is not already one.

Each time a user tries to access a system, he indicates an account a and types the target string. While the user is typing, *keystroke data* is captured, and a *sample* is created containing the *features* calculated using this data. If the account is new (enrollment), then the training set is collected, and a *template* is created containing the patterns found in it. A sample will only be stored if the *key code* feature matches. If the account already exists (authentication), then a sample will be analyzed by the *classifier* using the account's template to decide if the sample belongs to the account's owner. If the sample is considered true, then the user could access the system. Otherwise, another trial is conceded. In this second trial, if the classifier decides again that the sample is false, then the user is considered an impostor. This way, a session of authentication could be in three situations: one successful trial, a failed first trial and a successful second one, or a two failed trials. Finally, an *adaptation* mechanism could be activated to compute a new template update.

The main issues related to the methodology are described in the following sections.

A. Timing Accuracy

The basic foundation for the typing biometrics is to have an accurate and reliable data source of typing patterns in time [14]. In this work, the Time Stamp Counter function was used to catch the count of clock cycles as in [14]. The Time Stamp Counter keeps an accurate count of every cycle that occurs in the processor [15].

The precision adopted has to be compatible with the range values of the collected samples. Since 98% of the collected samples' values are between 10 and 900 ms, a 1-ms precision was used in this paper.

B. Keystroke Data

A string with m characters will result in n keystrokes, where $m \leq n$, since some characters need more than one keystroke. The keystrokes used in the sample w account a are represented by $K_{a,w} = \{k_1(a,w), k_2(a,w), \ldots, k_n(a,w)\}$.

Each keystroke $k_i(a,w)$ for $i \leq n$ is composed of the key down time $t_{i\cdot\mathrm{down}}(a,w)$ (the instant when the key is pressed), the key up time $t_{i\cdot\mathrm{up}}(a,w)$ (the instant when the key is released) and the key code $c_i(a,w)$ (the ASCII code).

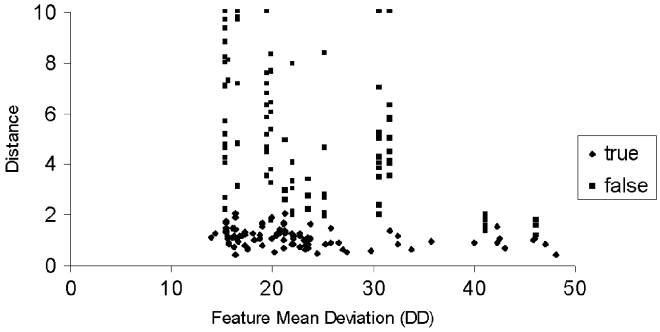

Fig. 2. DD separation between true samples and false samples.

C. Features

Features are calculated using keystroke data. Four features were analyzed in this paper: *key code* and three time-features (*down-down* (DD), *down-up* (DU), and *up-down* (UD) times). The key code is a novel feature. The other features had already been analyzed in previous studies, but only applying at most two of them.

1) Key Code: Key code is the ASCII code that represents each key in a keyboard. When a string contains capital letters, there are more than one possible set of key codes, otherwise there is a single one.

$C_a = \{c_1(a), c_2(a), \ldots, c_n(a)\}$ denotes the key codes contained in the template of the account a and $C_{a,w} = \{c_1(a,w), c_2(a,w), \ldots, c_n(a,w)\}$ denotes the key codes contained in the sample w in the account a.

2) Down-Down Time: DD time is a keystroke latency defined as the time interval between successive keystrokes [5]. This feature is represented by $\mathrm{DD}_{a,w} = \{dd_1(a,w), dd_2(a,w), \ldots, dd_n(a,w)\}$, where $dd_i(a,w) = t_{i+1\cdot\mathrm{down}}(a,w) - t_{i\cdot\mathrm{down}}(a,w)$ is related to (k_i, k_{i+1}).

Fig. 2 shows the DD separation between genuine samples (provided by account's owners) and false samples (provided by impostors) according to the feature mean deviation and the distance between a sample and the template. This distance will be explained in the classifier session.

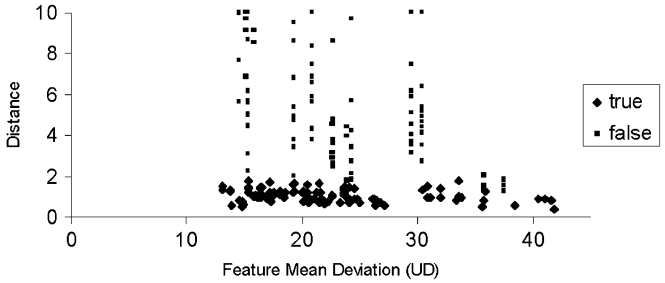

Fig. 3. UD separation between true samples and false samples.

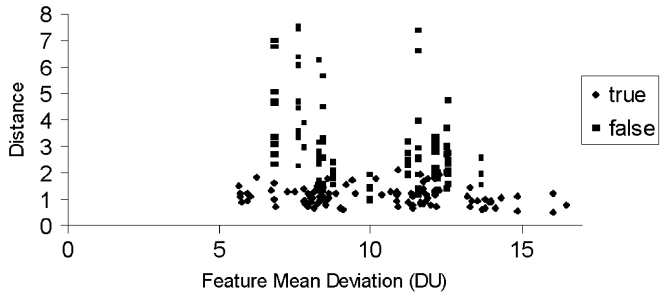

Fig. 4. DU separation between true samples and false samples.

3) Up-Down Time: UD time is also a keystroke latency feature represented by $\mathrm{UD}_{a,w} = \{ud_1(a,w), ud_2(a,w), \ldots, ud_{n-1}(a,w)\}$, where $ud_i(a,w) = t_{i+1\cdot\mathrm{down}}(a,w) - t_{i\cdot\mathrm{up}}(a,w)$ is related to (k_i, k_{i+1}). Fig. 3 shows the UD separation between the true samples and the false samples.

This feature could be positive or negative according to two situations. In the first situation, k_{i+1} is only pressed when k_i was released which results in a positive time value. In a second situation, k_{i+1} is pressed while k_i is still pressed, which results in a negative time value.

4) Down-Up Time: The DU time is defined as the time interval that a key remains pressed [5]. This feature is represented by the expression $\mathrm{DU}_{a,w} = \{du_1(a,w), du_2(a,w), \ldots, du_n(a,w)\}$, where $du_i(a,w) = t_{i\cdot\mathrm{up}}(a,w) - t_{i\cdot\mathrm{down}}(a,w)$ is related to k_i. Fig. 4 shows the DU separation between the true and false samples.

As seen in Figs. 2–4, DD and UD result in a better separation compared with DU.

D. Template

The template contains the key code itself C_a and the mean μ and the standard deviation σ that are calculated for each element i of each feature *feat* (DD, DU, or UD) through its jth sample of the training set $j \leq 10$ according to (1) and (2), as follows:

$$\mu_{\mathrm{feat}_i(a)} = \frac{1}{10}\sum_{j=1}^{10} \mathrm{feat}_i(a,j) \tag{1}$$

$$\sigma_{\mathrm{feat}_i(a)} = \frac{1}{10-1}\sum_{j=1}^{10} \left|\mathrm{feat}_i(a,j) - \mu_{\mathrm{feat}_i(a)}\right|. \tag{2}$$

E. Classifier

The sample w of the account a is analyzed by the classifier. Initially, $C_{a,w}$ is compared with C_a: if they are different, the sample is considered false; otherwise, for each time feature, the distance in standard deviation units between the template and the sample is calculated by (3)

$$D_{\mathrm{feat}}(a,w) = \frac{1}{n}\sum_{i=1}^{n} d_i(a,w) \tag{3}$$

where n is the number of elements of the feature $feat$ and $d_i(a,w)$ is the distance related to each element i between the template and the sample, and is given by (4)

$$d_i(a,w) = \frac{\mathrm{feat}_i(a,w) - \mu_{\mathrm{feat}_i(a)}}{\sigma_{\mathrm{feat}_i(a)}}. \tag{4}$$

Finally, the sample will be considered true if the condition $D_{\mathrm{dd}}(a,w) \leq T_{\mathrm{dd}}(a)$ and $D_{\mathrm{du}}(a,w) \leq T_{\mathrm{du}}(a)$ and $D_{\mathrm{ud}}(a,w) \leq T_{\mathrm{ud}}(a)$ is satisfied, where $T_{\mathrm{dd}}(a), T_{\mathrm{du}}(a)$, and $T_{\mathrm{ud}}(a)$ are the thresholds for the DD, DU, and UD features, respectively, in the account a.

The determination of the threshold is an important issue in the methodology. Therefore, an analysis was performed in real collected keystroke data, and it was observed that a user's feature with a higher variation demands a lower threshold, meanwhile a user's feature with a lower variation demands a higher threshold. So, the threshold for each feature in each account is obtained based on its standard deviation.

F. Adaptation Mechanism

The adaptation mechanism consists of creating a new updated template, including the new sample and discarding the oldest one. This mechanism is performed after a successful authentication with a sample w if the majority elements i of its time features *feat* satisfy the following condition $(d_{\mathrm{feat}_i}(a,w) \leq T_{\mathrm{feat}}(a))$. As the adaptation mechanism is performed, the standard deviation for each feature is modified and the thresholds are modified.

IV. Experiments

The experiments were conducted on three machines with two different keyboard layouts. Thirty users (men and women between 20 and 60 years old) participated in the experiments in three situations of authentication.

- *Legitimate user authentication*: the users tried to be authenticated in their own account. Each legitimate user tried to be authenticated between 15 and 20 times, which resulted in 553 collected sessions.
- *Impostor user authentication*: the users tried to be authenticated in other user's accounts, knowing the string typed by their owners. Each account was attacked by these users between 80 and 120 times, which resulted in a total of 2916 collected sessions.
- *Observer impostor user authentication*: the users observed how the other users type their strings, then they tried to be authenticated in their accounts. Each account was attacked by these users between 12 and 20 times, which resulted in a total of 492 collected sessions.

These samples were collected in the Laboratory of Pattern Recognition and Computer Networks (LRPRC) in UNICAMP/Brazil. Each user's samples were collected in different periods of time, never all at once.

Seven experiments were analyzed, combining the features: 1) only DD time; 2) only UD time; 3) only DU time; 4) DD and UD times; 5) DD and DU times; 6) UD and DU times; 7) DD, UD, and DU times.

TABLE I
COMPARATIVE RESULTS IN EXPERIMENT FOR LEGITIMATE USERS

Experiment	Sessions	Errors	FRR
(I)	553	9	1.63%
(II)	553	12	2.17%
(III)	553	13	2.35%
(IV)	553	12	2.17%
(V)	553	7	1.27%
(VI)	553	9	1.63%
(VII)	**553**	**8**	**1.45%**

TABLE II
COMPARATIVE RESULTS IN EXPERIMENT FOR IMPOSTOR USERS

Experiment	Sessions	Errors	FAR
(I)	2916	580	19.90%
(II)	2916	179	6.14%
(III)	2916	795	27.26%
(IV)	2916	151	5.18%
(V)	2916	163	5.59%
(VI)	2916	91	3.12%
(VII)	**2916**	**55**	**1.89%**

TABLE III
COMPARATIVE RESULTS IN EXPERIMENT FOR OBSERVER IMPOSTOR USERS

Experiment	Sessions	Errors	FAR
(I)	492	144	29.27%
(II)	492	58	11.79%
(III)	492	166	33.74%
(IV)	492	46	9.35%
(V)	492	48	9.75%
(VI)	492	34	6.91%
(VII)	**492**	**18**	**3.66%**

A. Results

The performance of biometrics systems are generally measured by two kinds of error rates [16].

- *False Acceptance Rate (FAR)*: the probability that the system will fail to reject an impostor user.
- *False Rejection Rate (FRR)*: the probability that the system will fail to verify the legitimate user claimed identity.

Other performance measures based on these rates are [1] as follows.

- *ZeroFAR*: FRR when the FAR is equal to zero.
- *ZeroFRR*: FAR when the FRR is equal to zero.
- *Equal Error Rate (EER)*: the value when the FAR and FRR are equally likely.

Table I shows the FRR in each experiment for legitimate users. Tables II and III show the FAR in each experiment for impostor users and observer impostor users, respectively.

As noted in Tables I and II, the best results were achieved in the experiment (VII). In this experiment, 1.45% FRR and 1.89% FAR were obtained. The FAR increased in Table III to 3.66% when using the observer impostor's samples. Fig. 5 shows the Receiver Operating Characteristics (ROC) curve achieved in the experiment (VII) and points out the ZeroFAR (14.3%), ZeroFRR (12.2%), and EER (1.6%).

The operating threshold employed by a system depends on the nature of the application and it is very difficult to find a system that operates in one of these three points [1]. In practical applications, the system is configured to operate around or between these points.

B. Discussion

The following observations were made according to some experiments realized.

- As observed in Tables II and III, the FAR increases from 1.89% using simple impostor users' samples to 3.66% using observer impostor users' samples. This way, even if an impostor observes a legitimate user typing the string, this does not mean that the impostor will be authenticated.
- The choice of a target string with capital letters, which combines shift and Caps Lock keys, increases the difficulty of authentication of an impostor user. Since, with this kind of target string, 70% of impostor's sessions were detected using only this feature.
- The familiarity of the target string to the user has a significant impact. A 17.26% FRR was achieved changing the chosen string typed by the imposed one "unicamp2003."
- One-trial authentication significantly increases the FRR. A 11.57% FRR was achieved conferring just one trial in each session.
- The adaptation mechanism decreases both rates. A 4.70% FAR and a 4.16% FRR were achieved without the adaptation mechanism.
- If the adaptation mechanism is always activated, the FAR increases a lot, since impostor user samples are used to update the user's template. A 9.4% FAR and a 3.8% FRR were achieved in this experiment.
- A higher timing accuracy decreases both rates. A 1.63% FRR and a 3.97% FAR were achieved in a lower timing accuracy.
- FRR increases as the number of samples is reduced. Meanwhile, the FAR decreases as the number of samples is reduced, but not significantly. Fig. 6 shows the influence of the number of samples used in the enrollment.

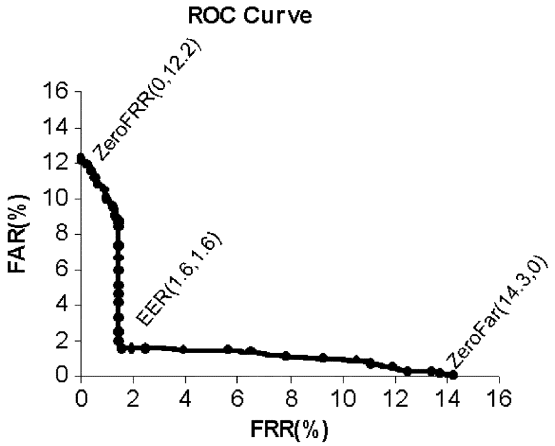

Fig. 5. ROC curve achieved in the experiment (VII).

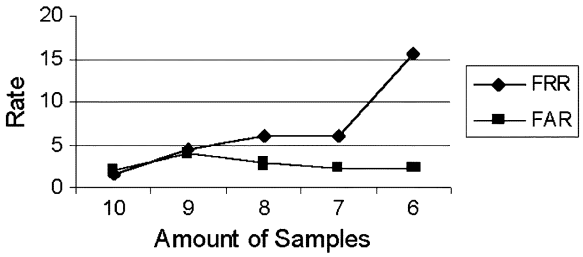

Fig. 6. Variation of the FAR and FRR rates with the number of samples.

Table IV shows a summary of results of some previous studies on keystroke dynamics, including a previous one based on fuzzy logic conducted by us and presented in [9]. In [12], both rates, 2% FRR and 6% FAR, are higher than ours, 1.45% FRR and 1.89% FAR, and they used 15 samples. The 13.3% FRR achieved in [3] is almost equivalent to a ZeroFAR (0.17% FAR), which is almost equal to ours (14.3% ZeroFAR). However, they used four different strings. In [10], both rates, 7.4% FRR and 2.8% FAR, are higher than ours, and they used two strings. Finally, in [9], our work using fuzzy logic approach and eight samples obtained a 3.5% FRR and a 2.9% FAR that higher to our new rates. Therefore, our research is competitive to other studies published in the same area.

TABLE IV
COMPARISON OF OUR APPROACH WITH SOME PREVIOUS STUDIES

Research	Samples	String	FRR	FAR
De Ru and Elof [10]	Varies	Two	7.4%	2.8%
Joyce and Gupta [3]	Eight	Four	13.3%	0.17%
Haidar et al. [12]	Fifteen	One	2.0%	6.0%
Araujo et al. [9]	Eight	One	3.5%	2.9%
Our Research	Ten	One	1.45%	1.89%

V. CONCLUSION

This paper presents a methodology through typing biometrics features that improves the usual login-password authentication. Some experiments were conducted and the best performance was achieved using a statistical classifier based on distance and the combination of four features (key code, DD, UD, and DU times), obtaining a 1.45% FRR and a 1.89% FAR. These rates, as discussed in the experiments section, are both competitive if compared with previous studies, using just one target string and ten samples in enrollment. The use of four features to authenticate users is novel, since prior studies used just one or two features.

This paper shows the influence of some practical aspects, which were tested and observed and shows that they have a relevant influence in the performance results. These aspects are: the familiarity of the target string, the two-trial authentication, the adaptation mechanism, the timing accuracy, and the number of samples in enrollment.

For future work, we intend to increase our user population and to extend the methodology to numeric keyboard used in access control of restricted areas and in banking transactions.

REFERENCES

[1] D. Polemi. (1997) Biometric Techniques: Review and Evaluation of Biometric Techniques for Identification and Authentication, Including an Appraisal of the Areas Where They are Most Applicable. Institute of Communication and Computer Systems, National Technical University of Athens, Athens, Greece. [Online]. Available: ftp://ftp.cordis.lu/pub/infosec/docs/biomet.doc, EU Commission Final Rep.

[2] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Gen. Comput. Syst.*, vol. 16, no. 4, pp. 351–359, 2000.

[3] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *Commun. ACM*, vol. 33, no. 2, pp. 168–176, 1990.

[4] D. Bleha and M. Obaidat, "Dimensionality reduction and feature extraction applications in identifying computer users," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 2, pp. 452–456, Mar.–Apr. 1991.

[5] D. T. Lin, "Computer-access authentication with neural network based keystroke identity verification," in *Proc. Int. Conf. Neural Networks*, vol. 1, 1997, pp. 174–178.

[6] S. Bleha, C. Slivinsky, and B. Hussain, "Computer-access security systems using keystroke dynamics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 12, pp. 1217–1222, Dec. 1990.

[7] J. A. Robinson, V. M. Liang, J. A. Michael, and C. L. MacKenzie, "Computer user verification login string keystroke dynamics," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, no. 2, pp. 236–241, Mar.–Apr. 1998.

[8] M. S. Obaidat and B. Sadoun, "Verification of computer user using keystroke dynamics," *IEEE Trans. Syst., Man, Cybern.*, vol. 27, no. 2, pp. 261–269, Mar.–Apr. 1997.

[9] L. C. F. Araújo, L. H. R. Sucupira Jr., M. G. Lizárraga, L. L. Ling, and J. B. T. Yabu-uti, "A fuzzy logic approach in typing biometrics user authentication," in *Proc. 1st Indian Int. Conf. Artificial Intelligence*, 2003, pp. 1038–1051.

[10] W. G. de Ru and J. H. P. Eloff, "Enhanced password authentication through fuzzy logic," *IEEE Expert*, vol. 17, no. 6, pp. 38–45, Nov.–Dec. 1997.

[11] O. Coltell, J. M. Badfa, and G. Torres, "Biometric identification system based in keyboard filtering," in *Proc. IEE 33rd Annu. Int. Carnahan Conf. Security Technology*, 1999, pp. 203–209.

[12] S. Haidar, A. Abbas, and A. K. Zaidi, "A multi-technique approach for user identification through keystroke dynamics," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, vol. 2, 2000, pp. 1336–1341.

[13] F. Monrose, M. K. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," in *Proc. 6th ACM Conf. Computer Security*, Singapore, Nov. 1999.

[14] F. W. M. H. Wong, A. S. M. Supian, A. F. Ismail, L. W. Kin, and O. C. Soon, "Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm," in *Conf. Rec. 35th Asilomar Conf. Signals, Syst., Comput.*, vol. 2, 2001, pp. 911–915.

[15] Using the RDTSC Instruction for Performance Monitoring (1997). [Online]. Available: http://developer.intel.com/drg/pentiumII/appnotes/rdtscpm1.htm

[16] A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*. Norwell, MA: Kluwer, 1999.