Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2008, Article ID 345047, 14 pages
doi:10.1155/2008/345047

Research Article

Design and Evaluation of a Pressure-Based Typing Biometric Authentication System

Wasil Elsadig Eltahir,[1] M. J. E. Salami,[1] Ahmad Faris Ismail,[1] and Weng Kin Lai[2]

[1] *Faculty of Engineering, International Islamic University (IIUM), Jalan Gombak, Kuala Lumpur 53100, Malaysia*
[2] *Centre for Advanced Informatics, MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur 57000, Malaysia*

Correspondence should be addressed to Wasil Elsadig Eltahir, d.jwasel@gmail.com

Received 2 July 2007; Revised 3 November 2007; Accepted 16 May 2008

Recommended by C. Vielhauer

The design and preliminary evaluation of a pressure sensor-based typing biometrics authentication system (PBAS) is discussed in this paper. This involves the integration of pressure sensors, signal processing circuit, and data acquisition devices to generate waveforms, which when concatenated, produce a pattern for the typed password. The system generates two templates for typed passwords. First template is for the force applied on each password key pressed. The second template is for latency of the password keys. These templates are analyzed using two classifiers. Autoregressive (AR) classifier is used to authenticate the pressure template. Latency classifier is used to authenticate the latency template. Authentication is complete by matching the results of these classifiers concurrently. The proposed system has been implemented by constructing users' database patterns which are later matched to the biometric patterns entered by each user, thereby enabling the system to accept or reject the user. Experiments have been conducted to test the performance of the overall PBAS system and results obtained showed that this proposed system is reliable with many potential applications for computer security.

1. INTRODUCTION

Although a variety of authentication devices to verify a user's identity are in use today for computer access control, passwords have been and probably would remain the preferred method. Password authentication is an inexpensive and familiar paradigm that most operating systems support. However, this method is vulnerable to intruder access. This is largely due to the wrongful use of passwords by many users and to the unabated simplicity of the mechanism. This simplicity makes such system susceptible to unsubstantiated intruder attacks. Methods are needed, therefore, to extend, enhance, or reinforce existing password authentication techniques.

There are two possible approaches to achieve this, namely by measuring the time between consecutive keystrokes "latency" or measuring the force applied on each keystroke. The pressure-based biometric authentication system (PBAS) has been designed to combine these two approaches so as to enhance computer security.

PBAS employs force sensors to measure the exact amount of force a user exerts while typing. Signal processing is then carried out to construct a waveform pattern for the password entered. In addition to the force, PBAS measures the actual timing traces "latency." The combination of both information "force pattern and latency" is used for the biometric analysis of the user.

As compared to conventional keystroke biometric authentication systems, PBAS has employed a new approach by constructing a waveform pattern for the keystroke password. This pattern provides a more dynamic and consistent biometric characteristics of the user. It also eliminates the security threat posed by breaching the system through online network as the access to the system is only possible through the pressure sensor reinforced keyboard "biokeyboard".

Figure 1 shows PBAS block diagram. The operation of the system relies on constructing a users' database and then processing this information online through data classifiers.

The database stores users' login names, passwords, and biometric patterns. Data classifiers are used to analyze and

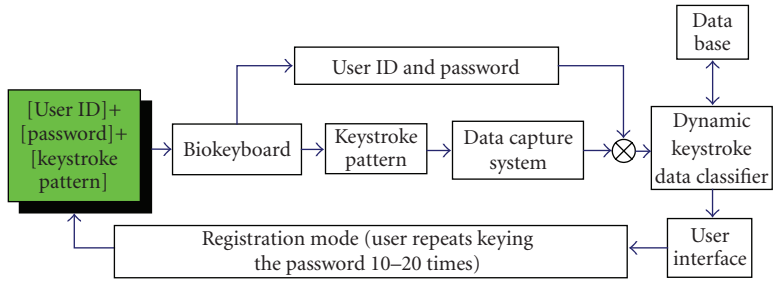

Figure 1: PBAS block diagram.

associate users with distinctive typing characteristic models. PBAS has been tested with combination of two classifiers, namely:

(1) autoregressive classifiers,

(2) latency classifiers.

These classifiers have been tested and the results obtained from the experimental setup have shown that these classifiers are very consistent and reliable.

2. DESIGN OF PRESSURE-BASED TYPING BIOMETRIC AUTHENTICATION SYSTEM (PBAS)

Keystroke authentication systems available in the market are mostly software-based. This is due to the ease of use as well as the low cost of the mechanism. Any new keystroke authentication system has to consider these factors in the design. Likewise, the system designed for PBAS uses simplified hardware which minimizes the cost of production. The system is designed to be compatible with any type of PC. Moreover, it does not require any external power supply. In general, the system components are low cost and commonly available in the market.

The operation of the system is depicted in Figure 1. System starts by prompting user to enter his/her user ID and password. The alphanumeric keyboard (biokeyboard) extracts the pressure template for the password entered. At the same time, the system calculates the latency pairs for the entered password and accompanies it with pressure template in a single data file. This data file is transferred to the system's database.

In the learning mode, the user is required to repeatedly key in the password for several times (10–20) to stabilize his/her keystroke template.

In the authentication mode, the user is requested to enter his/her ID and password. The resulting pressure template and latency vector are compared with those modeled in the database using the AR and latency classifiers. Depending on the results of this comparison, the user will be either granted or denied access to the system.

2.1. System hardware components

As illustrated in Figure 2, the main hardware components of PBAS are as follows:

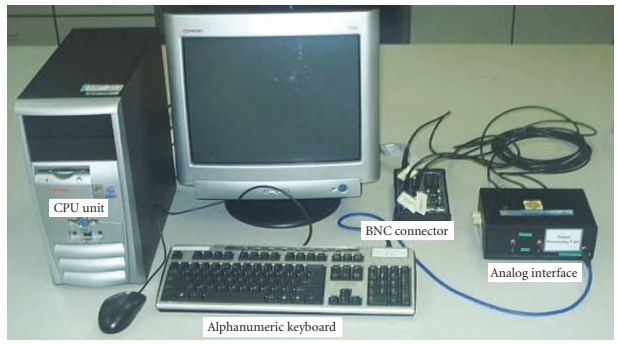

Figure 2: Integration of PBAS components.

(1) alphanumeric keyboard (biokeyboard) embedded with force sensors to measure the keystroke pressure while typing;

(2) data acquisition system consisting of the following components:

(a) analog interface box (filtering and amplification of signal),

(b) DAQ PCI card fitted into the PC.

(3) PC/central processing unit (CPU) for running the PBAS program using Windows XP operating system.

2.2. Pressure sensitive alphanumeric keyboard (biokeyboard)

A special keyboard was manufactured to acquire the alphanumeric password and the keystroke pressure template of the user. The biokeyboard layout is identical to normal commercial keyboard. This is crucial to maintain an intrinsic system that does not alter user typing habits. Figure 3 shows the biokeyboard front, back, and side views.

To measure the keystroke pressure, ultra thin flexible force sensors are fixed below each keyboard key. A plastic spring is fixed between the key and the sensing area to ensure that it does not get dislodged. This is necessary to avoid erroneous readings.

The keyboard operates just as a normal alphanumeric keyboard in addition to measuring keystroke pressure. Thus, the users of this system would not find any differences between this keyboard and the commercial ones.

Figure 3: Pressure sensitive alphanumeric keyboard (biokeyboard).

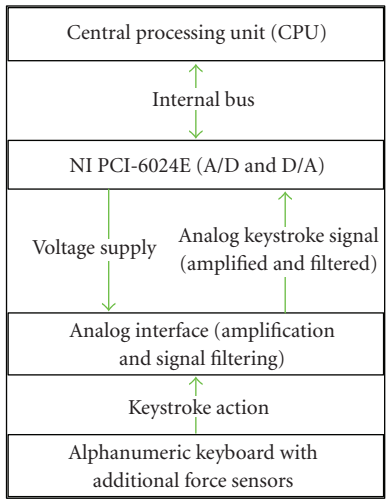

Figure 4: Operation of data acquisition system.

2.3. Data acquisition system

The force sensors are connected in parallel and then to the sensor drive circuit. The drive circuit is contained inside the analogue interface box (see Figure 2). The connection between the keyboard and the analogue interface box is made through a cable. Figure 4 shows the connection and operation of the data acquisition system.

The analogue interface box passes the keystroke pressure template from the biokeyboard to the PC through the DAQ PCI card. It contains amplification and filtering circuit to improve the voltage acquired from the biokeyboard. The analogue interface box also contains two knobs to adjust the sensitivity of the voltage (and hence keystroke pattern) by changing the amplification gain of the drive circuit.

Some further signal processing procedures are used to concatenate keystroke signals of different keys pressed when typing a password. This concatenation forms a continuous pattern for each keystroke password.

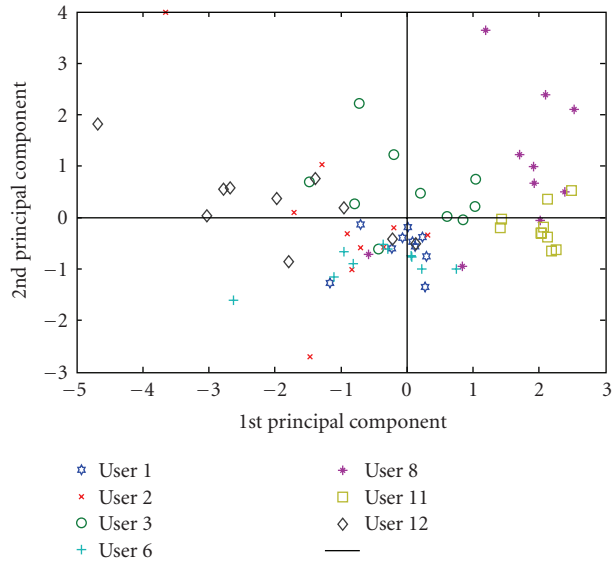

Figure 5: PCA for latency.

2.4. Validation of keystroke force approach

An experiment has been conducted to evaluate the significance of force analysis in the classification of users keystroke typing biometrics. In this experiment a group of 12 professional typists were asked to type a common password *tri-msn4*. The system acquired the latency and peak force for each character of the password entered by users. Each subject was required to type the same password 10 times. Here, each typed password consists of seven latency and eight peak force features, resulting in fifteen features for each user.

Principle component analysis (PCA) was then applied to analyze the dataset over first two dominant principal components axis. Three different classification cases were examined, namely: (a) classification by latency, (b) classification by peak force, (c) and lastly classification by combining latency and peak force.

Latency features were similar as seen in Figure 5. This is logical for consistent typists because they use the same hand and wrist lateral positions when typing and hence they tend to type with almost the same speed.

The results in Figure 5 show that users 11 and 8 have distinctive latencies while users 1, 3, and 6 exhibit high similarities that can be considered as a group. User 12 on the other hand has a relatively high variation.

In Figure 6 it is apparent that peak force has better classification as compared to that of latency. This is justified by the fact that the typing force varies for different typists. However, similarities amongst each single user's data points are somehow lower than that of latency. Thus, we conclude that keystroke force is comparatively higher in variation than latency.

As may be seen in Figure 7, combining force and latency has improved the data classification for the users. This diagram illustrates that data clustering of each single user is better with the combined analysis of force and latency. Since the two variables vary in different manners, it is therefore

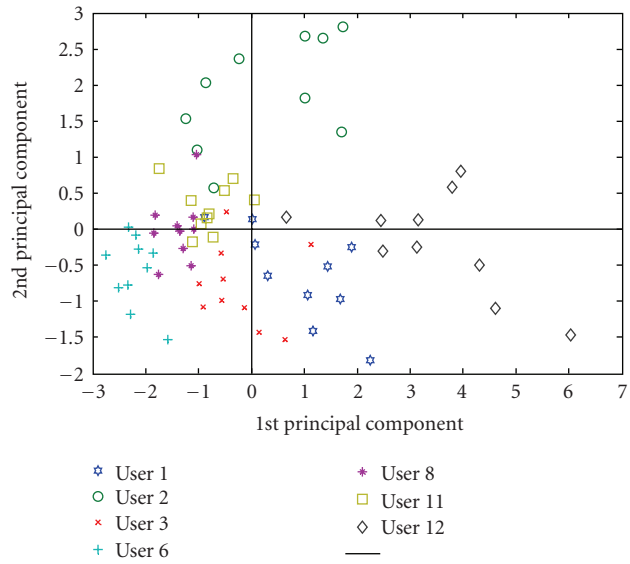

Figure 6: PCA for peak force.

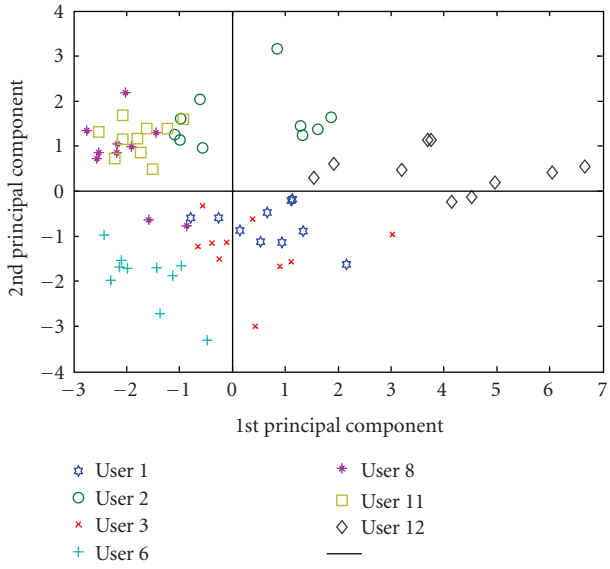

Figure 7: PCA for latency and peak force.

necessary to design two classifiers to measure (or evaluate) them.

3. DYNAMIC KEYSTROKE CLASSIFIERS

Dynamic keystroke template results from a distinctive keystroke action. When a user enters a password, a single keystroke is applied on each key pressed.

Figure 8 shows a typical pressure template acquired for a password of six characters. The template is for user "MJE1" and the password used is "123asd."

This diagram shows that the pressure template points are interrelated in time and are of random nature. This would

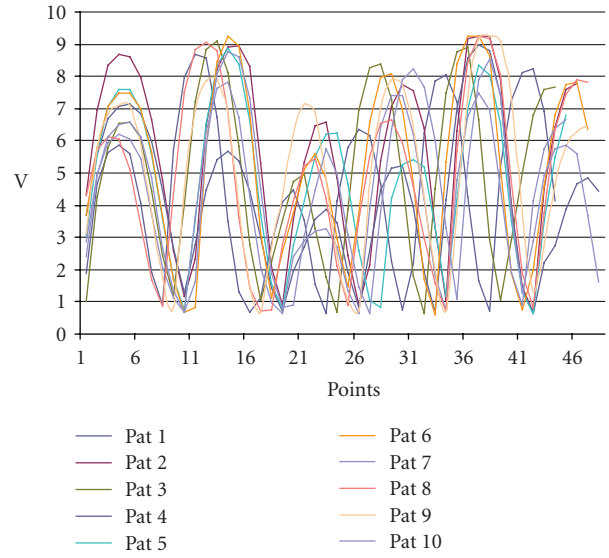

Figure 8: Keystroke pattern for single-user six-character password.

suggest that statistical signal analysis may be useful to classify these templates.

AR classifier based on stochastic signal modeling has been developed for the classification of the keystroke pressure template. As for the keystroke latency, a separate classifier has been developed based on the key down action. This classifier is used together with the AR-based keystroke pressure classifier. These classifiers are discussed in detail in the following sections.

3.1. *Latency classifier*

Keystroke authentication using time digraphs (latency) has been investigated thoroughly with many researchers [6–10]. Many useful methodologies have been presented and are in use with the current latency keystroke authentication systems available in the market.

Joyce and Gupta discussed the design of identity verifier based on four input strings (login name, password, 1st name, and last name). The verification is done by comparing the mean reference signature "M" with a test signature "T." The norm $\|M - T\|$ is computed and if this norm is less than the threshold for the user, the attempt is accepted; otherwise it is flagged as an imposter attempt [7].

Though this approach produces relatively satisfactory results, it requires relatively lengthy input string. A modified approach has been devised here for PBAS latency authentication. PBAS uses the password string only for latency verification.

3.1.1. *Creating mean reference latency vector*

(1) Registered users are prompt to reenter their password (10–20) times, latency vector for each trial is saved in an individual data file resulting in (n) number of files in the database, where n is the number of trials.

(2) Data treatment is applied on the data files to remove outliers and erroneous values.

(3) An average latency vector is calculated using the user trial sample. This results in a single file containing the mean latency vector (R) for n password trials. This file is used as reference which will be used for latency authentication.

3.1.2. Calculating suitable threshold

Thresholding is used to decide an acceptable difference margin between the reference latency vector (R) and the latency vector provided by the user upon verification (V). The threshold is computed based on the data files saved in the database. A threshold is set for each user based on the variability of his latency signatures. A user that has little variability in his latencies would have a small threshold. User with high variability should have larger threshold. Standard deviation is the variability measure used.

Standard deviation between the mean (R) latency vector and the user sample is measured. A threshold based on the standard deviation is used for authentication based on the following rule:

$$\sum_{k=1}^{m-1} |R_k - V_k| \leq c*d, \tag{1}$$

where m is the password length, R_k is the kth latency value in the reference latency vector, V_k is the kth latency value in the user-inputted latency vector, c is an access threshold that depends on the variability of the user latency vector, and d is the distance in standard deviation units between the reference and sample latency vectors.

In order to classify user attempt, we define the latency score S_L for the user attempt to be

$$S_L = \frac{\sum_{k=1}^{m-1} |R_k - V_k|}{c*d}. \tag{2}$$

Therefore, depending on the value of S_L, the classifier output will be

$$S_L \begin{cases} \leq 1, & \text{accept template,} \\ > 1, & \text{reject template.} \end{cases} \tag{3}$$

Table 1 shows the reference latency vector for user "MJE1" which was calculated by the above mentioned method for a sample of 10 trials. Five latency vectors are used to test the threshold c for this reference profile (see Table 1). The standard deviation was calculated to be $S_y = 46.5357$ milliseconds and a threshold of *2 standard deviations above the mean* ($c = 2$) resulted in the following variation interval $253.9748 \geq R - V \geq 67.83189$. This threshold takes in all 5 trials of the user. However, this is a relatively high threshold value and in many practical situations such values would only be recommended for unprofessional users who are usually not very keen typists. The user here is a moderate typist. This is evident by his relatively high standard deviation. High standard deviation is also a measure of high variability in the users' latency pattern; this usually indicates that the user template has not yet stabilized, perhaps due to insufficient training.

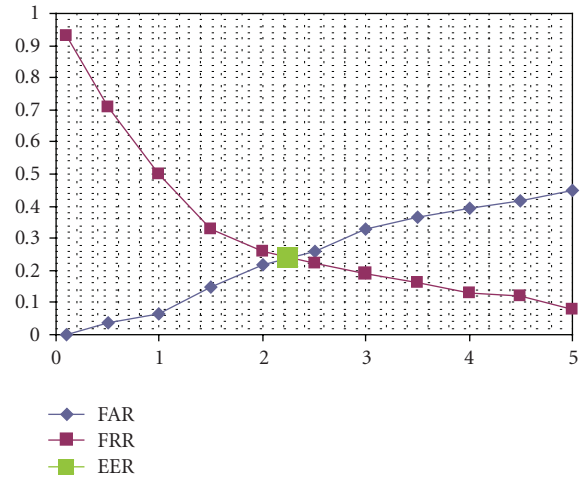

Figure 9: Latency threshold versus FAR and FRR rates.

Table 2 shows the variation of threshold values c (from 0.5 to 2.0) and its effect on accepting the user trials.

For this user, a threshold value that is based on standard deviation of 2.0 provides an acceptance rate of 100% (after eliminating outliers). However, a high threshold value would obviously increase the imposter pass rate. Therefore for normal typists, the threshold values should only be within the range of 0.5 to 1.5.

An experiment was conducted to assess the effect of varying the latency threshold value on the FAR and FRR rates. In this experiment, an ensemble for 23 authentic users and around 50 intruders were selected randomly to produce authentic and intruder access trials. Authentic users were given 10 trials each and intruders were given 3 trials per account. All trials were used for the calculations and no outliers were removed. The graphical user interface used was normal (see Figure 18). Figure 9 shows that the equal error rate (EER) for the FAR and the FRR rates was 24% and it occurred at a threshold value of 2.25. This relatively high FAR rate is expected since the password strings used were mainly short in length and weak in strength.

3.2. AR-Burg classifier

The AR algorithm uses the notion of signal analysis to reproduce the users' keystroke pressure template. The reproduced template is then compared with the keystroke template produced by the alleged intruders. Based on this comparison an authentication decision is made.

A signal model approach is advocated here since the pressure template points are interrelated across time. The AR signal model is defined as follows:

$$y(n) + a(1)y(n-1) + a(2)y(n-2) + \cdots + a(p)y(n-p) = x(n), \tag{4}$$

Table 1: Reference latency tested against 5 authentic user trials.

Password (asd123)	Reference latency vector	T 1	T 2	T 3	T 4	T 5
a-s	201.9231	203	187	172	203	235
s-d	162.8571	156	188	171	156	93
d-1	316.2308	235	187	219	188	250
1-2	185.2000	187	203	203	187	203
2-3	108.5714	110	110	110	94	79

Table 2: Effect of threshold value on user acceptance rate.

Standard deviation	Upper limit	Lower limit	Acceptance percentage
0.5	184.1712	137.6355	40
1.0	207.439	114.3676	60
1.5	230.7069	91.09975	80
2.0	253.9748	67.83189	100

where n is the time index, $y(n)$ is the output, $x(n)$ is the input, and p is the model order.

For signal modeling $y(n)$ becomes the signal to be modeled and the $a(i)$ coefficients need to be estimated based on the signal's characteristics.

If we use the above equation to predict future values of the signal $\hat{y}(n)$, the equation becomes

$$\hat{y}(n) = -a(1)y(n-1) - a(2)y(n-2) - \cdots - a(p)y(n-p). \tag{5}$$

Now, we define the error from $e(n)$ to be the difference between the predicted and the actual signal point. Therefore $e(n)$ can be defined as

$$y(n) + a(1)y(n-1) + a(2)y(n-2) + \cdots + a(p)y(n-p) = e(n). \tag{6}$$

The total squared error (TSE) for predicted signal is

$$\text{TSE} = \sum_{n=1}^{N-1} e_n^2. \tag{7}$$

The AR model is used most often because the solution equations for its parameters are simpler and more developed than those of either moving average (MA) or autoregressive moving average (ARMA) models [1, 2].

Burg method has been chosen for this application because it utilizes both forward and backward prediction errors for finding model coefficients. It produces models at lower variance (S_p^2) as compared to other methods [1].

Authentication is done by comparing the total squared error TSE percentage of the users in the database with that generated by the linear prediction model. Previous experiments proved that authentic users can achieve TSE margin of less than 10% [3].

3.2.1. Identifying optimum pressure template for AR modeling

An algorithm was developed in Matlab to identify the best pressure template in the user sample. This pattern is used for estimating the AR model parameters of the user keystroke pressure. The algorithm uses the correlation technique to calculate the accumulative correlation index (ACI) which is the accumulation of the correlation between each pressure pattern and the whole sample. The pattern with the highest ACI is chosen for the model.

3.2.2. Identifying the optimum TSE acceptance margin

The TSE relative prediction error (RPE) is calculated by the following equation:

$$\text{Relative\_Prediction\_Error} = \left| \frac{\text{TSE}_m - \text{TSE}_s}{\text{TSE}_m} \right|, \tag{8}$$

where TSE_m is the TSE calculated for the user's AR-Burg model in database. TSE_s is the TSE for the pressure pattern of the user.

Classification of user attempt is done by comparing RPE to threshold T according to the following:

$$\text{RPE} \begin{cases} \leq T, & \text{accept template,} \\ > T, & \text{reject template,} \end{cases} \quad \text{where, } 0 < T \leq 1. \tag{9}$$

Based on previous research experiments [3], it was reported that authentic users can achieve up to 0.1 RPE while intruders exhibit unbounded fluctuating RPE that can reach above 3.0 [3].

An experiment was conducted to assess the effect of varying the TSE threshold value on the FAR and FRR rates. In the experiment, an ensemble for 23 authentic users and around 50 intruders were selected randomly to produce authentic and intruder access trials. Authentic users were given 10 trials each and intruders were given 3 trials per account. All trials were used for the calculation of results and no outliers were removed. The graphical user interface used was normal (see Figure 18). Figure 10 shows how the FAR and the FRR vary as we change the TSE threshold values. The EER was 25% and it was recorded at TSE of 37.5%. Compared to latency, TSE has lower FRR spread out as the threshold is increased.

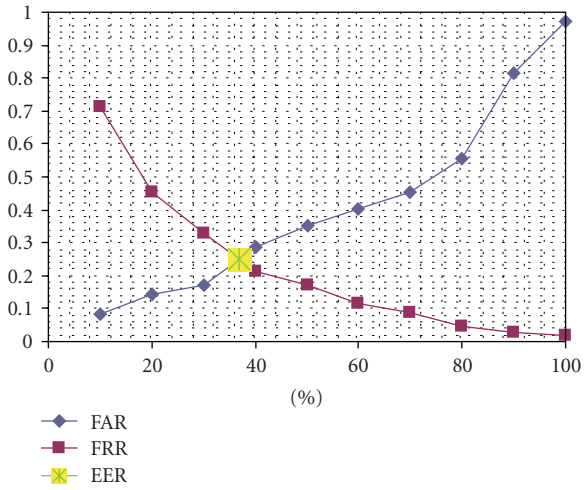

Figure 10: TSE threshold versus FAR and FRR rates.

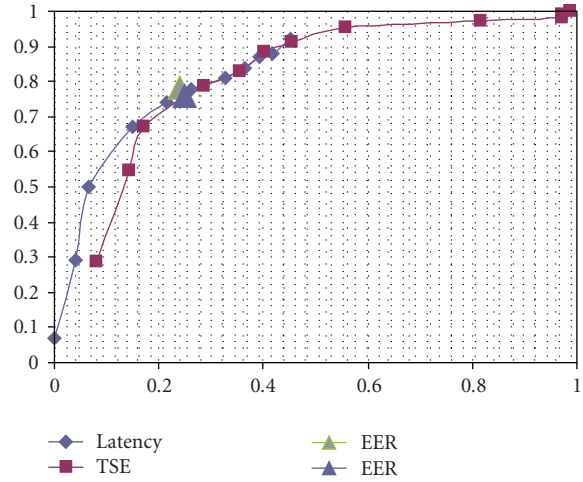

Figure 11: ROC showing performance of latency and TSE classifiers.

The AR modeling algorithm has been implemented in the following order.

(1) The user is prompted to enter the password several times (20 times).

(2) The optimum pattern for modeling the user is identified using the ACI values obtained from the sample.

(3) The best AR model order is determined based on the *final prediction error* (FPE) and the *Akaike's information criteria* (AIC).

(4) The AR model is constructed and model coefficients are saved for user verification.

(5) Using AR model coefficients, the linear prediction model is constructed to predict the original template from the pattern entered by the user.

(6) Using the linear prediction model TSE_m is calculated for user's template in database. The RPE score is used to discriminate between authentic and intruder attempts.

(7) If RPE $\leq T$, user is authentic, whereas if RPE $> T$, then user is intruder.

3.3. *Receiver operating curve for TSE and latency classifiers*

The receiver operating characteristic curve (ROC) is used to assess the effect of the threshold value on the FAR and FRR rates. ROC curve assesses the trade-off between low intruder pass rate and high authentic pass rate as the decision threshold value varies. Figure 11 shows that the latency classifier has slightly better separation than the AR classifier. In addition to that, the latency classifier has better intruder rejection rate whereas AR classifier has a higher true pass rate. The graph also shows that the performance of both classifiers at the EER points is very similar; therefore, it is expected that by combining both algorithms the overall system performance will be improved. The operating range for the AR classifier is between 0.1 and 1.0 threshold values of T corresponding to very low FAR and FRR rates, respectively. The operating range for the latency classifier is between 0.1 and 5.0 threshold values c corresponding to very low FAR and FRR rates, respectively.

4. SYSTEM ALGORITHMS AND PROGRAM STRUCTURES

With the integration of software and hardware, the PBAS algorithm was designed to have two main operation modes.

(1) Training users and creating biometric template profiles; at this stage the user is requested to key in his/her ID and the user trains his/her password.

(2) Authenticating existing users based on the identity they claim; users provide ID and password which are compared with the biometric profiles of the users in the database.

Figure 12 shows the flow graph for the overall PBAS training and authentication process. The authentication mode consists of two phases.

(1) Normal authentication, which involves the password combination and its compliance with the one saved in the database.

(2) Biometric authentication, which is done by the combination of latency along with the AR classifiers.

Firstly, the user will select the mode of operation. In the training mode, the access-control system requests the user to type in the login ID and a new password. The system then asks the user to reenter the password several times in order to stabilize his/her typing pattern. The resulting latency and pressure keystroke templates are saved in the database. During training, if the user mistypes the password the system prompts user to reenter the password from

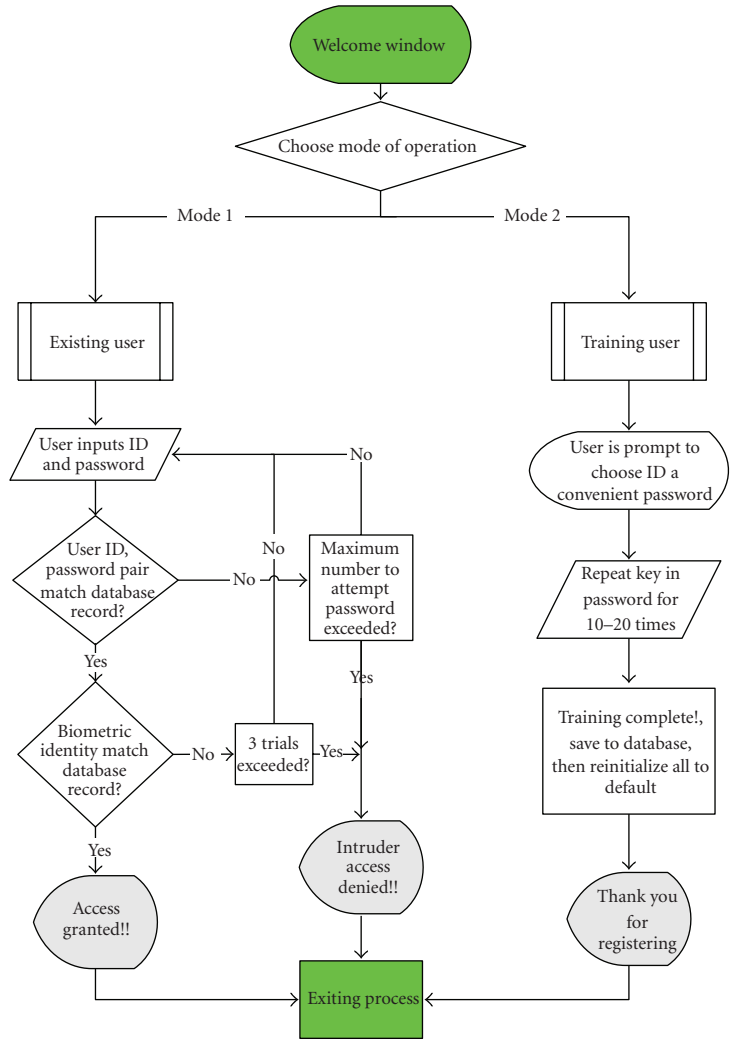

Figure 12: PBAS main algorithm flowchart.

the beginning. The use of backspace key is not allowed as it disrupts the biometric pattern. When registration is done, system administrator uses these training samples to model user keystroke profiles. The design of user profiles is done offline. After that, the administrator saves the users' keystroke template models along with the associated user ID and password in the access-control database.

In the authentication mode, the access-control system requests the user to type in the login ID and a password. Upon entering this information the system compares the alphanumeric password combination with the information in the database. If the password does not match, the system will reject the user instantly and without authenticating his keystroke pattern. However, if the password matches then the user keystroke template will be calculated and verified with the information saved in the database. If the keystroke template matches the template saved in database, the user is granted access.

If the user ID and alphanumeric password are correct, but the new typing template does not match the reference template, the security system has several options, which can be revised occasionally. A typical scenario might be that PBAS advises a security or network administrator that the typing pattern for a user ID and password is not authentic and that a security breach might be possible. The security administrator can then closely monitor the session to ensure that the user does nothing unauthorized or illegal.

Another practical situation applies to automatic teller machine (ATM) system. If the user's password is correct but the keystroke pattern does not match, the system can restrict the amount of cash withdrawn on that occasion to minimize any damages made by possible theft or robbery.

5. EXPERIMENTS ON PBAS PERFORMANCE USING COMBINED LATENCY AND AR CLASSIFIERS

As concluded from the ROC curve (Figure 11), it is expected that combining the latency and TSE classifiers will produce better authentication results. The threshold used for the TSE classifier will be $T = 0.4$ as recommended by the

Figure 13: Choosing mode of operation.

Figure 14: Registering new user.

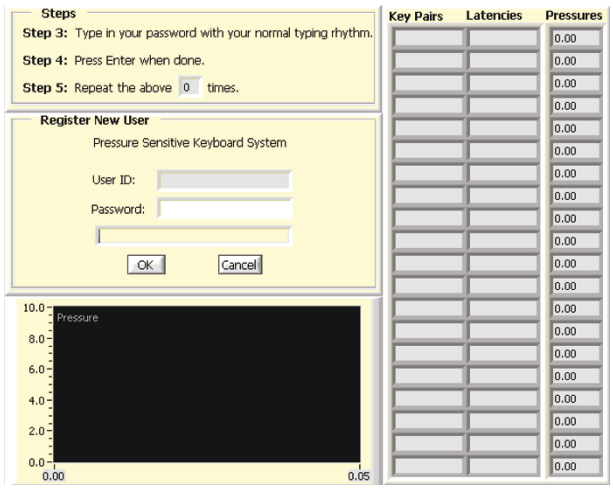

Figure 15: Training user password.

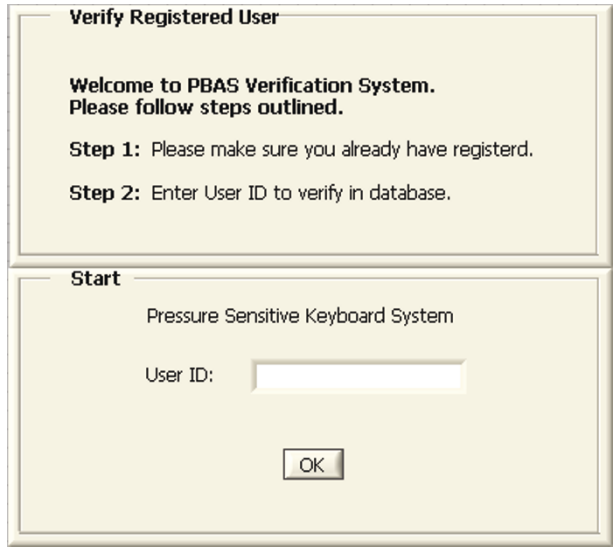

Figure 16: Validation mode.

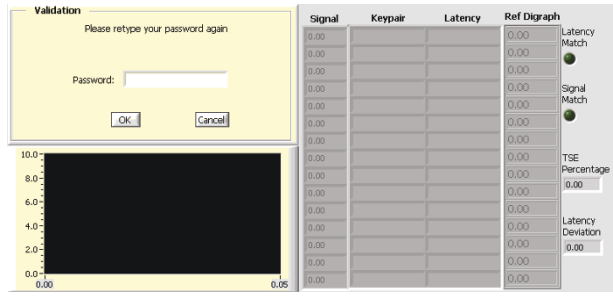

Figure 17: Guided validation (experiment 1).

EER calculated earlier. As for the latency threshold c, it is recommended to use a threshold value between 2.0 and 2.25 for unprofessional typists and 1.0 to 1.5 for professional typists.

5.1. Experimental variables

The experimental variables that are assumed to play role in the performance of the system are as follows: (1-) user disposition, (2-) intruder knowledge of the authentic keystroke latency, (3-) sensitivity of the data acquisition system, (4-) strength of user password, (5-) sampling rate of the data acquisition system, and (6-) threshold values of the AR and latency classifiers.

In the experiment, four variables were fixed and two were varied for analysis. These variables were

(1) intruder knowledge of the authentic keystroke latency,

(2) threshold values of the AR classifier.

By varying these two variables, we will be able to answer two important questions.

(1) How does exposing the database to intruders affect the system security?

(2) What is the effect of increasing the TSE percentage on the FAR rate?

The following section will try to answer these questions. In addition, we will try to analyze the user passwords and identify possible reasons behind any successful intruder attacks.

Two experiments were conducted with a population of 23 users. Eleven of the participants were females and 12 were males. Participants were of different ages (18 to 50). One participant "user3" was left handed. Training and authentication for each user password were done on two different occasions (at least not on the same day).

All users participating in the experiments were briefed thoroughly about the operation of PBAS. They were also told about the purpose of the experiment to ensure maximum interaction from users.

At the beginning, users were asked to choose an ID and password, "ID up to eight characters and password not less than six characters". The users trained their password for twenty trials. The administrator created AR-keystroke model and latency vector for each user and saved it in the system database.

All 23 users participated in the first experiment. However, only successful hackers were inducted to the second experiment.

In both experiments, a simple program with interactive GUI would first ask the user to key in his/her ID, and then the computer would create a random list of 10 accounts "five male and five female" for the user to attempt hacking.

To calculate the FAR in both experiments, users were asked to repeat keying the password for 10 times. The results were evaluated online by recording the instances of acceptance and rejection for each user.

5.2. Experimental procedure

The two experiments were arranged as follows.

5.2.1. Experiment 1: "guided authentication"

In this experiment hackers were allowed to see the users' reference latency vector along with their own pressure template, a GUI window was fixed with two indicator lights "one for latency and one for pressure" that flashes green when either latency or pressure is within the acceptance margin. TSE threshold T was set to 0.15.

Authentic users were given ten attempts per account whereas intruders were given four hacking attempts per account. Twenty three registered users participated in this experiment generating a total of 230 authentic attempts, 19 of these users participated as intruders generating a total of 760 intruder attacks. According to Figure 10, it is expected that the FRR will be as high as 60% and that the FAR will be as low as 11% (knowing that the tests are different).

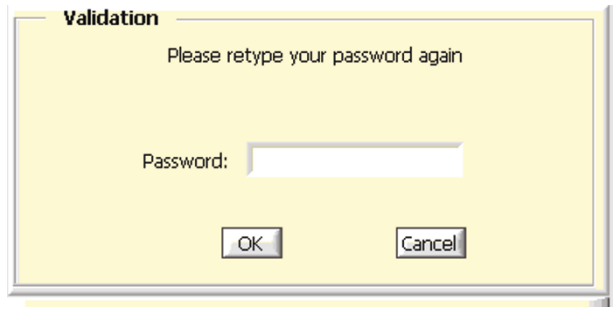

Figure 18: Normal validation (experiment 2).

5.2.2. Experiment 2: "normal authentication"

In this experiment GUI window was restricted not to show any information about user pressure or latency vectors. RPE threshold T was set to 0.4; this increase was made to reduce the FRR rate as recommended from the ROC curve (Figure 11). Authentic users were given 10 attempts whereas intruders were given 3 hacking attempts per account. All 23 authentic users participated in this experiment generating a total of 230 authentic attempts. As for the intruder attempts, only 8 users "successful hackers of experiment 1" participated in this experiment generating a total of 240 intruder attacks. According to Figure 10, it is expected that the FRR will be as high as 21% and that the FAR will be as low as 28% (knowing that the tests are different).

5.3. Experimental results

While the computer security society recommends that a safe password should be a combination of alphabets, numeric, and special characters, almost 80% of users have chosen passwords that do not conform to the standard measures of password safety. Some users chose their login ID as the password; some used standard words, combination of repeated letters, or combination of adjacent keyboard keys with no special characters. All of these factors have rendered the users' passwords very vulnerable with respect to the password security standards. Our assumption is that PBAS will improve the performance of weak passwords by combining the latency and AR classifiers. Table 4 shows the results for the experiments conducted.

The FRR for the first experiment was 10.43% which is very much less than the maximum expected FRR of 60%. This could be attributed to the improved typing efficiency of the users which minimizes the occurrence of outliers during the experiment.

It is noticed that the increase in AR threshold T from 0.15 to 0.4 has reduced the FRR by 70% while increasing the FAR by 138%.

Table 5 shows the cross comparison for the FRR rate recorded for the 8 successful hackers across experiments 1 and 2. The table shows that the increase in the AR threshold T along with the removal of feedback did not increase the

Table 3: FAR and FRR for experiments 1 and 2.

User ID	Password	User attempts	Intruder attempts	Experiment 1		Intruder attempts	Experiment 2	
				FAR%	FRR%		FAR%	FRR%
$*^{1}$ + ^{1}User 1(m)	123asd	10	44	**2.272**	**10**	**15**	**6.666**	**10**
User 2(m)	dadn4tay	10	52	**0**	**0**	**15**	**0**	**0**
User 3(m)	alshaijiy	10	48	**0**	**20**	**12**	**0**	**0**
$*^{1}$ User 4(m)	mmu123	10	64	**1.562**	**20**	**3**	**0**	**10**
$*^{1}$ User 5(m)	tigna12	10	48	**2.083**	**10**	**15**	**0**	**0**
User 6(m)	mohammedhasan44	10	48	**0**	**0**	**9**	**0**	**0**
$*^{2}$ User 7(f)	tastas13	10	48	**4.166**	**0**	**9**	**0**	**0**
User 8(f)	dowrita-	10	52	**0**	**20**	**9**	**0**	**10**
User 9(m)	nanana	10	20	**0**	**20**	**6**	**0**	**20**
User 10(m)	lrtvib	10	56	**0**	**0**	**3**	**0**	**0**
User 11(m)	obahja1313	10	32	**0**	**0**	**12**	**0**	**0**
User 12(m)	sal12sal	10	36	**0**	**20**	**15**	**0**	**0**
User 13(m)	alrubataby	10	20	**0**	**0**	**12**	**0**	**0**
User 14(m)	hussam44	10	20	**0**	**0**	**6**	**0**	**0**
$*^{3}$ +^{3} User 15(f)	fardusa	10	44	**6.818**	**0**	**15**	**20**	**0**
User 16(f)	salah1	10	48	**0**	**10**	**6**	**0**	**0**
User 17(f)	ktwon123	10	48	**0**	**10**	**15**	**0**	**0**
$*^{2}$ User 18(f)	alkontabbad	10	44	**4.545**	**30**	**3**	**0**	**10**
User 19(f)	suli00	10	32	**0**	**0**	**15**	**0**	**0**
User 20(f)	asma23	10	48	**0**	**10**	**15**	**0**	**0**
$*^{1}$ +^{4} User 21(f)	fathiya	10	32	**3.125**	**20**	**18**	**22.22**	**0**
$*^{1}$ +^{1} User 22(f)	faduma	10	20	**5**	**20**	**12**	**8.333**	**10**
User 23(f)	subway	10	16	**0**	**20**	**6**	**0**	**0**

*Denotes hacked logins in 1st experiment, superscript is number of hacks.
+Denotes hacked logins in 2nd experiment, superscript is number of hacks.

Table 4: Total FAR and FRR for experiments 1 and 2.

Experiment	Number of authentic participants	Number of intruder participants	Number of attacks	Number of successful attacks	FAR%	FRR%
I	23	19	760	12	1.57	10.43
II	23	8	240	9	3.75	3.04

Table 5: Comparing FRR for successful hackers in experiments 1 and 2.

Experiment	Number of intruder participants	Number of attacks	Number of successful attacks	FAR%
I	8	320	12	3.75
II	8	240	9	3.75

FAR rate; this means that the removal of feedback canceled the effect of increasing T threshold. Hence, there is some correlation between knowledge of the verifier and the ability of an imposter to match the reference signature of another user.

Table 6 shows a comparison between results obtained here and previous research efforts. A comparison is not statistically valid as these systems use different sample size with different parameters and methodologies to measure the keystroke. It is important to note that earlier research emphasized on the strength of the password string and as a result, the users had to use either lengthy strings (sometimes 4 strings) or strong strings (combination of alphanumeric keys and special characters). PBAS, however, does not require lengthy or strong password strings. Consequently, it is more user friendly, but on the other hand this makes it more susceptible to intruder attacks.

5.3.1. *Statistical significance of experimental results*

It is important to assess the statistical significance of the results obtained in this experiment. In general statistics, the larger the number of volunteers and the number of attempts made (sample size), the more accurate the results would be [4].

Table 6: Comparison of our results with previous efforts.

Research	Number of participants	Training samples	Password string	FAR%	FRR%
Legget and Williams (1988) [6]	36	12	large	5.00	5.50
Joyce and Gupta [7]	33	8	4	13.30	0.17
De Ru and Eloff [8]	29	Varies (2 to 10)	1	2.80	7.40
Haider et al. [9]	Not mentioned	15	1	6.00	2.00
Araújo et al. [10]	30	10	1	1.89	1.45
Our research	23	20	1	3.75	3.04

To calculate the variance for the FRR rate we use the following:

$$\hat{p} = \frac{1}{n}\sum p_i = \frac{1}{mn}\sum a_i,$$
$$\hat{V}(\hat{p}) = \frac{\sum (p_i - \hat{p})^2}{n(n-1)} = \frac{1}{(n-1)}\left[\frac{\sum a_i^2}{m^2 n} - \hat{p}^2\right], \tag{10}$$

where n is the number of enrolled volunteers; m is the average number of samples per volunteer; a_i is the number of false nonmatches for the ith volunteer; $p_i = a_i/m_i$ is the proportion of unmatched samples for the ith volunteer; $\hat{p}$ is the observed FRR for all volunteers; $\hat{V}(\hat{p})$ is the estimated variance of the observed FRR rate.

For experiment 2, FRR = 0.0304. The variance was calculated to be 1.357×10^{-4}.

To find the 95% confidence interval, we substitute for the variance in the following:

$$\hat{p} \pm z(1 - \alpha/2)\sqrt{\hat{V}(\hat{p})}, \tag{11}$$

where $z()$ is the area under standard normal curve with mean zero. For 95% confidence, $z(0.975)$ is 1.96. The 95% confidence interval for the true error rate (p) is $0.0075 \leq p \leq 0.0532$.

To calculate the confidence interval for the FAR rate, we use the following [5].

If the product $N * \hat{p} \geq 10$, (where N is number of independent trials and $\hat{p}$ is the observed FAR rate) then we may use the normal distribution curve to approximate the 95% confidence interval as follows:

$$\hat{p} - 2\sigma_{\hat{p}} \leq p \leq \hat{p} + 2\sigma_{\hat{p}}, \tag{12}$$

where p is the true FAR rate, $\sigma_{\hat{p}}$ is the maximum likelihood estimator which is defined as

$$\sigma_{\hat{p}} = \frac{1}{N}\sqrt{e\left(1 - \frac{e}{N}\right)}, \tag{13}$$

where e is the number of successful intruder attacks.

The estimated FAR rate recorded for experiment was 0.0375 the 95% confidence interval for the true FRR rate is calculated as follows:

$$0.01626 \leq p \leq 0.05844. \tag{14}$$

5.3.2. Recommendations on test size

To improve the statistical significance and accuracy of our results, we recommend the following.

(1) Firstly, the number of enrolled users should be increased to at least 100 users.

(2) Then, collect 15 genuine samples per user to produce a total of 1500 genuine samples. This is above the requirement of the rule of 30.

(3) Use cross comparison with 10 users per intruder attack allowing 3 trials per attack. This will produce 3000 intruder attacks. This is above the requirement of 30.

(4) To minimize the dependency of the intruder attacks by the same person, it is recommended to collect these data in two sessions.

(5) Finally, once the data has been collected and analyzed, the uncertainty in the observed error rates would be estimated in order to ascertain the size of the test data.

5.4. Discussion of results

The following observations can be inferred from Table 3.

(i) Since the computer-generated attack list was random, the number of intruder attacks per user account was variable. Nevertheless, all accounts have been tested for intrusion.

(ii) Users who chose passwords identical to their user name (user 15, 21, and 22) suffered highest rate of successful intruder attacks.

(iii) Users 1, 4, 7, and 9 had substantially weak passwords. As expected, users 1, 4, and 7 were susceptible to successful intruder attacks. However, user 7 repelled all intruder attacks and after investigation, it was found that user 7 had a highly distinctive keystroke pressure template.

(iv) Users who chose standard passwords that comply with security measures achieved maximum protection and were able to better resist intruder attacks.

(v) In the experiment, there was one left-handed user, "user3." His keystroke pressure template was strong against intruder attacks. Investigations showed that

while right-handed users exerted more pressure typing on right side keys of the keyboard, this left-handed user exerted more pressure on the left side keys and hence his pressure template was distinctive from right-handed intruders.

(vi) The decrease in latency threshold reduces the FRR rate.

(vii) Users with low latency standard deviation were able to better repel intruder attacks. This is logical since low standard deviation suggests that the users typing pattern is more stable and hence the corresponding pressure template almost match that of the user.

(viii) The increase of AR threshold from 0.15 to 0.40 has decreased the FRR rate significantly.

(ix) The intruder knowledge has some effect on his ability to succeed in attacking other user accounts.

6. CONCLUSION

In the course of the last 30 years, keystroke biometrics has emerged as a quick and user-friendly solution for access control systems.

Several commercial keystroke biometric algorithms have been introduced in the market.

However, most of these algorithms merely use the time information (latency) of the keystroke action and thus utilize only one information aspect of the keystroke action. It neglects the force applied in the keystroke action. PBAS has successfully acquired both information, time frame and applied force. Furthermore, the application of force (F) over time (Δt) has produced significant information in the form of a signal (pressure template); this approach is more dynamic and characteristic of user keystroke pattern.

Preliminary tests on PBAS indicated apparent success in the performance of the system. However, performance can be further enhanced to produce more accurate and reliable results.

Furthermore, the experiments have proved that force is highly distinctive to users typing keystroke. Although some users may have similar latency profiles, their keystroke pressure templates were easily discriminated. The reinforcement of pressure sensors to measure the keystroke force has many advantages such as:

(i) a password obtained by an imposter does not necessarily mean that the imposter can access the system;

(ii) a user's typing biometric is difficult to steal or imitate;

(iii) an imposter cannot obtain a user's typing biometrics by peeking at the user's typing;

(iv) the hardware reinforcement can be integrated to any password-based security system, because it works in conjunction with normal password mechanisms;

(v) the system administrator has the option of turning on/off the biometric reinforcement at anytime to use normal password authentication only.

Due to the fact that keystroke dynamics are affected by many external factors (position of hands while typing, fatigue, hand injuries, etc.), it is somehow difficult to ensure a typical pattern for a user's password every time. This inherent difficulty favors other biometric authentication techniques such as fingerprint and retina scan over keystroke biometrics. In order to overcome this difficulty, dynamic data classifiers are used with a suitable threshold to accommodate for the variability in user keystroke pattern.

The combination of AR and latency classifiers allows for an increase in latency threshold value to decrease the FRR rates. This increase does not have great effect on the FAR rates as the AR classifier would reject intruders based on their pressure templates, on the contrary it will make the system more user friendly without compromising the security.

AR technique uses AR-coefficients to reconstruct user pressure templates. This approach provides a more comprehensive user identity. Moreover, it is very easy to reconstruct the user pressure template for user authentication.

ACKNOWLEDGMENTS

The authors would like to thank IIUM and MIMOS for providing all the necessary resources to make this joint project successful. IIUM and MIMOS acknowledge the financial support from the Malaysian Ministry of Science, Technology and Innovation under Grant no. (IRPA-04-01-04-0006-EA 001) which has, in part, produced this paper.

REFERENCES

[1] R. Shiavi, *Introduction to Applied Statistical Signal Analysis*, Aksen Associates, Homewood, Ill, USA, 1991.

[2] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, New York, NY, USA, 1996.

[3] W. E. Eltahir, M. J. E. Salami, A. F. Ismail, and W. K. Lai, "Dynamic keystroke analysis using AR model," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '04)*, vol. 3, pp. 1555–1560, Hammamet, Tunisia, December 2004.

[4] A. J. Mansfield and J. L. Wayman, "Best practices in testing and reporting performance of biometric devices," Tech. Rep., National Physical Laboratory, Middlesex, UK, August 2002, version 2.01.

[5] J. E. Porter, "On the 30 error criterion," in *National Biometric Test Center Collected Works 1997–2000*, J. L. Wayman, Ed., pp. 51–56, National Biometric Test Center, San José State University, San Jose, Calif, USA, 2000.

[6] J. Leggett and G. Williams, "Verifying Identity via Keystroke Characteristics," *International Journal of Man-Machine Studies*, vol. 28, pp. 67–76, 1988.

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

[8] W. G. de Ru and J. H. P. Eloff, "Enhanced password authentication through fuzzy logic," *IEEE Expert*, vol. 12, no. 6, pp. 38–45, 1997.

[9] S. Haider, A. Abbas, and A. K. Zaidi, "A multi-technique approach for user identification through keystroke dynamics," in *Proceedings of the IEEE International Conference on Systems,*

Man and Cybernetics (SMC '00), vol. 2, pp. 1336–1341, Nashville, Tenn, USA, October 2000.

[10] L. C. F. Araújo, L. H. R. Sucupira Jr., M. G. Lizárraga, L. L. Ling, and J. B. T. Yabu-Uti, "User authentication through typing biometrics features," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, part 2, pp. 851–855, 2005.