Keystroke Dynamic Analysis Using Relative Entropy & Timing Sequence Euclidian Distance

H B Kekre, V A Bharadi
Computer Science Department
MPSTME, NMIMS University
Mumbai, India
+91-9323557897, +91-9819125676
hbk@yahoo.com,
vinu_bharadi@rediffmail.com

P Shaktia, V Shah
IT Department, TCET
Mumbai University
Mumbai, India
+91-7738343494
purva.shaktia@gmail.com,
vidicompany@yahoo.co.in

A A Ambardekar
Computer Science and Engineering,
University of Nevada
RENO, USA
+1-7753131894
amol_ambardekar@rediffmail.com

ABSTRACT

Biometric authentication systems are becoming more and more popular because of increased security concerns. Keystroke Dynamics is one of the important behavior based biometric trait. It has moderate uniqueness level and low user cooperation is required. In this paper keystroke dynamics analysis using relative entropy and Euclidian distance between keystroke timing sequence is discussed. In this approach keystroke timing sequence are calculated and normalized then this information is used for generating normalized probability distribution of dynamic passwords, two distance measures namely relative entropy & Euclidian distance are used for classification. This approach is simple and feasible for multimodal biometric systems.

Categories and Subject Descriptors

D.4.6 [Security and Protection]: Access Controls, Authentication

General Terms

Algorithms, Design, Experimentation, Security, Human Factors, Verification.

Keywords

Biometrics, Keystroke Dynamics, Relative Entropy.

1. INTRODUCTION

Biometrics systems are based on human behavior as well as physical characteristics. With advances in technology and programming platform faster computer, sensors are available and wide variety of programming platform make capturing and processing of biometric data easier and faster [1][2][3]. These factors have fuelled growth of biometrics market in last five years [**XXX**].

Biometric traits are mainly classified as physical traits like fingerprint, face, iris, palmprint or behavioral biometrics such as gait, handwritten signature, speech, keystroke dynamics etc. Depending of level of security, processing power, user population & implementation cost we can choose between different biometric traits or even more than one biometric traits can be combined to implement multimodal biometric systems [1], [2], [3]. In this paper keystroke dynamics based biometric authentication method is discussed. The way one user presses different keys on a keyboard is peculiar and unique. Along with the characters in a password this dynamic information can be used for identifying correct person.

1.1 Keystroke Dynamics

Keystroke dynamics, or typing dynamics, is the detailed timing information that describes exactly when each key was pressed and when it was released as a person is typing at a computer keyboard [271]-[274]. Keystroke dynamics is a safeguard based on authenticating access to computers by recognizing certain unique and habitual patterns in a user's typing rhythm. Unlike other behavior based biometrics keystroke dynamics are unique in that they do not need special sensor equipment; a normal keyboard can serve the purpose. The keystroke dynamics are captured entirely by application program through hardware driver, so the technique can be applied to any system that accepts and processes keyboard input events.

Keystroke dynamics can be used for single authentication events as well as continuous monitoring of keystroke events. Continuous monitoring has been proposed as a legitimate and reason able means to help prevent unauthorized use of unattended terminals, commercial products are still not widespread [1]. For example, if an authorized user leaves their computer terminal unattended and another user attempts to use it, the change in keystroke typing patterns could be recognized. The presence of the different person's keystrokes (which is by definition an unauthorized user or hacker in some environments) could then automatically generate a request for re-authentication. Continuous monitoring for the purposes of identification is a different procedure from keyboard logging or monitoring for auditing or eavesdropping purposes [1]. Keystroke monitoring is the primitive, yet considerably easy way to achieve logging of every key pressed by a user.

1.2 Existing Methods

Digraph & Trigraph based methods are explored by many researchers [1], [5], [7]. The keystroke timings are captured and then characters groups are formed. In digraph two characters are group and in trigraph three characters are grouped. Timing information of the groups formed to find degree of disorder [5] which is used to match the password sequences. Here we propose another metric for matching dynamic password sequences based

ICWET'11, February 25–26, 2011, Mumbai, Maharashtra, India.
Copyright © 2011 ACM 978-1-4503-0449-8/11/02…$10.00

on probability distribution function (PDF) generated from timing information.

It is argued that [4],[6] the use of keystroke rhythm is a natural choice for computer security. This argument stems from observations that similar neuro-physiological factors that make written signatures unique and they are also exhibited in a user's typing pattern [6]. When a person types, the latencies between successive keystrokes, keystroke durations, finger placement and applied pressure on the keys can be used to construct a unique signature (i.e., profile) for that individual. For well-known, regularly typed strings, such signatures can be quite consistent. Furthermore, recognition based on typing rhythm is not intrusive, making it quite applicable to computer access security as users will be typing at the keyboard anyway.

In this paper keystroke dynamic information extraction and matching is discussed. keystroke timing information is used to generate probability distribution for the password; this information is used for matching the password keystroke sequence containing timing information.

2. ENTROPY BASED MATCHING

In KD, there are two metrics used to verify the identity of a user: dwell time and flight time (see Figure B). As a person types, the KD application collects the duration of each key press and the cycle time between one key press and the next. Once the password is typed we calculate the information content of the password. This is done by calculating the entropy of the obtained timing values.

2.1 Entropy [4]

It is a measure of the uncertainty associated with a random variable. The term by itself in this context usually refers to the Shannon entropy, which quantifies, in the sense of an expected value, the information contained in a message, usually in units such as bits. The entropy H of a discrete random variable X with possible values {x1, ..., xn} is

$$H(X) = E(I(X)) \qquad (1)$$

Here 'E' is the expected value, and 'I' is the information content of X. I(X) is itself a random variable. If 'p' denotes the probability mass function of 'X' then the entropy can explicitly be written as

$$H(X) = \sum_{i=i}^{n} p(x_i) I(x_i) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i) \quad (2)$$

Where 'b' is the base of the logarithm used base=2.

For verification purposes a known verification string is usually typed (i.e. account ID and password). Once the verification string is entered, it is processed by an algorithm that compares the person's typing behavior to a sample collected in a previous session. The comparison is made by calculating the relative entropies of the sample and the test.

2.2 Relative Entropy [4]

Suppose the probabilities of a finite sequence of events is given by the probability distribution $P = \{p_1 ... p_n\}$, but somehow we mistakenly assumed it to be $Q = \{q_1 ... q_n\}$. According to this erroneous assumption, our uncertainty about the j^{th} event, or equivalently, the amount of information provided after observing the j^{th} event. The (assumed) average uncertainty of all possible events is then

$$-\sum_j p_j \log q_j \qquad (3)$$

On the other hand, the Shannon entropy of the probability distribution *p*, defined by,

$$-\sum_j p_j \log p_j \qquad (4)$$

is the real amount of uncertainty before observation. Therefore the difference between these two quantities

$$-\sum_j p_j \log q_j - \left(-\sum_j p_j \log p_j\right) = \sum_j p_j \log p_j - \sum_j p_j \log q_j \quad (5)$$

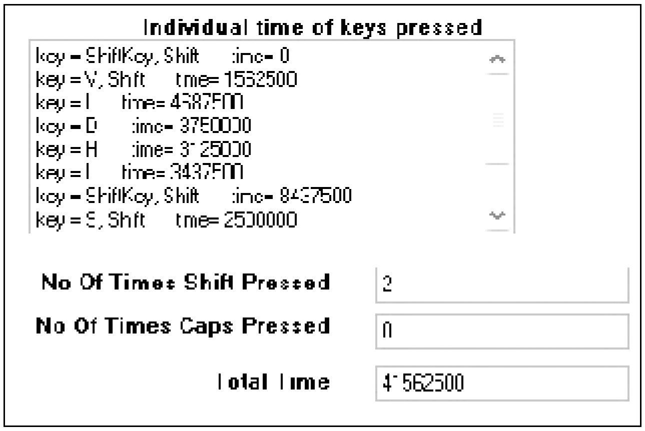
is a measure of the distinguishability of the two probability distributions *p* and *q*. This is precisely the classical relative entropy, or Kullback–Leibler divergence:

$$D_{KL}(P \parallel Q) = \sum_j p_j \log \frac{p_j}{q_j} \qquad (6)$$

The output of the comparison is a score. If this is the first time the KD system has seen this user, the results of this process are used to enroll him instead of verifying his identity. This distance has been used previously in matching wavelet energy sequence; here this metric is used for matching password keystroke's timing duration sequences.

3. CAPTURING KEYSTROKE'S TIMING INFORMATION

The password consists of characters. These characters are entered through keyboard, when a key is pressed an event is raised. Visual Studio 2008 is used for programming, in VS 2008 the related events are keydown, keyup & keypressed. We use these events to extract timing information of the password.



Individual time of keys pressed

```
key = ShiftKey, Shift    time= 0
key = V, Shift    time= 1562500
key = I    time= 4687500
key = D    time= 3750000
key = H    time= 3125000
key = I    time= 3437500
key = ShiftKey, Shift    time= 8437500
key = S, Shift    time= 2500000
```

No Of Times Shift Pressed: 2

No Of Times Caps Pressed: 0

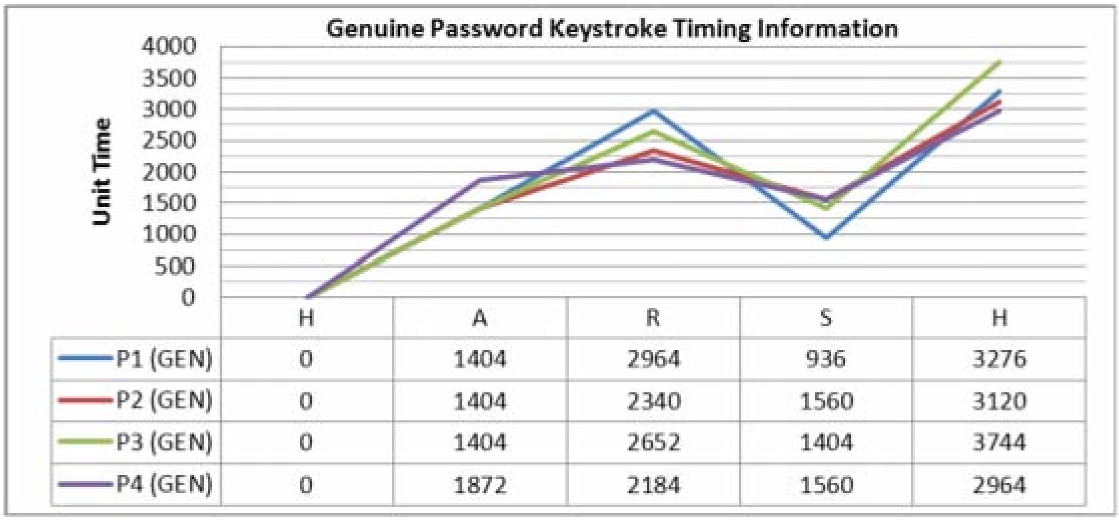
Total Time: 41562500

Figure 1. Captured Keystroke Data for password "VIDHIS"

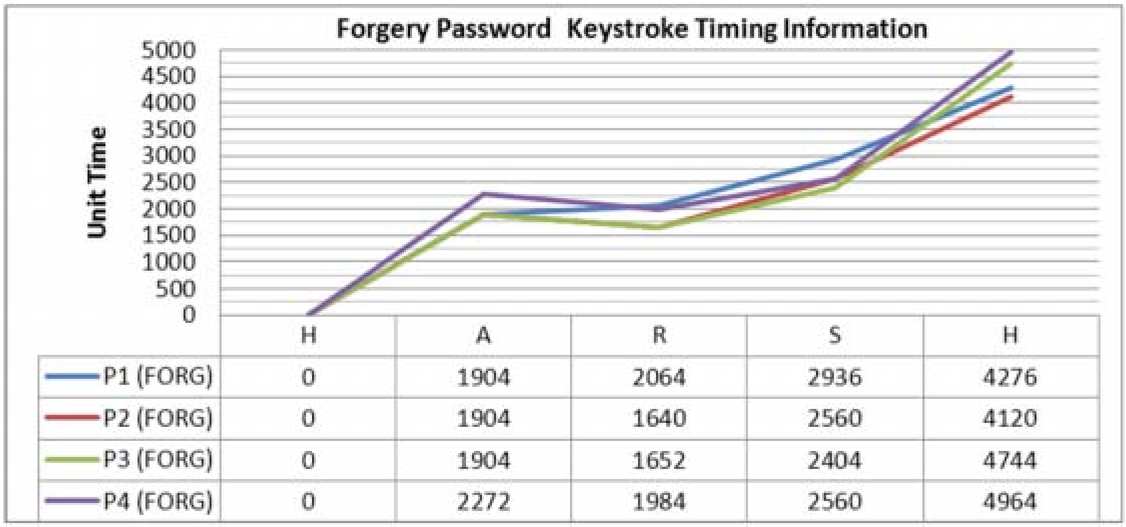
The captured time is in a composite format called as 'Ticks' in VS 2005 time units. The captured data is the time instance when the key is down, we get the keypress time & this data is normalized by converting into milliseconds & dividing each time value by total time. In this way timing information of different events 'E' is generated. E={E1,E2,,.….,En} , Where, Ei is ith key in the password.

Ei= (keypress Time)i + (keydown Time)i + (Flight Time)i (7)

The total time required to type the password is 'T' given by,



(a)



(b)

Figure 2. Captured Keystroke Data for Password "HARSH" Showing (SCALED) Plots for Genuine & Forged Password Keystroke Timings

$$T = \sum_{i=1}^{n} E_i \qquad (8)$$

From this information we generated the normalized PDF for the password 'P' as P= {P1, P2, P3, ….., Pn}.

$$P_i = \frac{E_i}{T} \qquad (9)$$

This normalized information is then used for password matching. The timing information for Genuine & Forged password (HARSH) is shown in Fig.2 (a) & (b).The passwords are scanned in one sitting hence it can be seen that the timing pattern for genuine & forgery is different. To match two password sequences Pi & Qi, Relative Entropy as discussed in this section as well as Euclidian distance between two timing sequences has been used.

4. RESULTS & DISCUSSIONS

This method is implemented using Microsoft Visual Studio 2008, & tested on Intel Core2Duo 2.4 GHz, 2GB RAM Running on Windows Vista Operating System. Ten passwords each from 33 different users were collected; the passwords were 4 to 6 characters long. For classifying passwords Euclidian distance between normalize PDF & relative entropy is used. Total 3325 Genuine & 1320 Forgery tests were performed for Distance Analysis.

Distance vs. its probability of occurrence is shown in Fig. 3., this plot is for the Euclidian distance between the password timing sequences. This clearly indicates that the password timing information can be used for classification. One more thing that should be noted is this is pure distance and not like 'Trigraph' & 'Bigraph' Sequences [5]. If we use this metrics then the separation is higher.

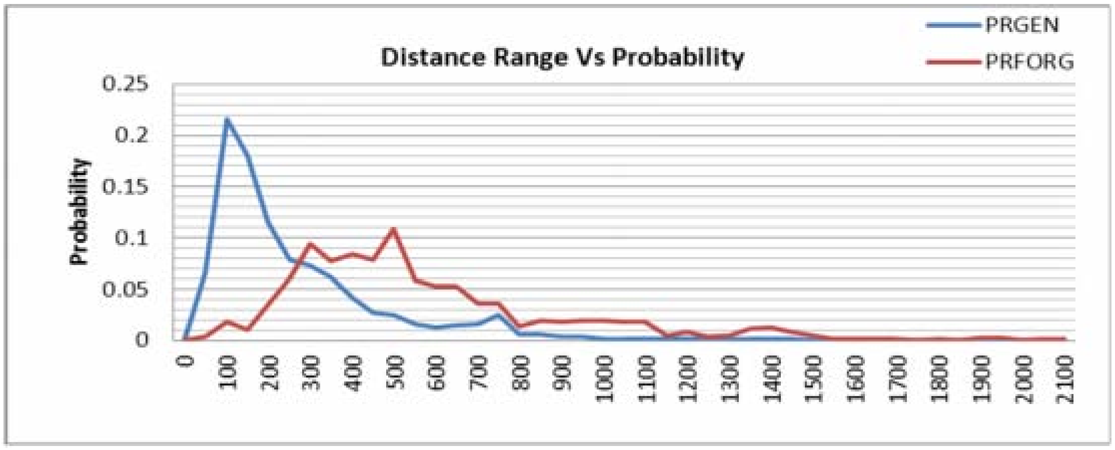
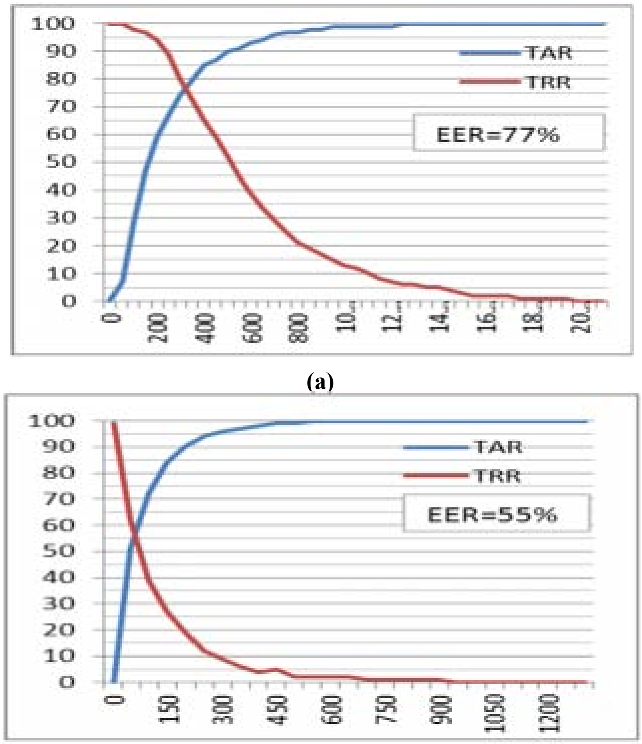


Figure 3. Distance Range vs. Probability. Two Peaks Corresponding to Genuine (PRGEN) & Forgery (PRFORG) are Clearly Visible.

TAR-TRR Analysis is shown in Fig. 4. (a) & (b). Euclidian distance based classifier performs better giving EER of 77%, Relative Entropy based classifier shows poor performance by giving EER=55%. Here simple K-NN classifier is used, with only 5 samples for training. The performance can be improved by considering longer password length of 6-8 characters and the use of special characters. The study shows the keystroke dynamics is viable option for computer security; it can be used to strengthen the password based authentication programs.



(a)

(b)

Figure 4. TAR-TRR Analysis for (a) Euclidian Distance Based Password Recognition (b) Relative Entropy Based Password Recognition.

5. CONCLUSION

In this another behavior based biometric 'keystroke Dynamics' is discussed. Euclidian distance between keystroke timing sequence as well as relative entropy between normalized passwords timing PDF is used for classification. The proposed approach is simpler and gives moderate accuracy. The accuracy can still be improved by implementing complex passwords and advanced classifiers based on neural networks. We can use this approach to build a multimodal biometric system using dynamic keystroke and other biometric trait. Currently the system is giving 77% accuracy in basic testing setup. The timing sequence PDF performs better as compared to entropy based matching. These metrics can be further investigated for keystroke dynamics analysis for performance improvement.

6. REFERENCES

[1] John D. Woodward, Jr. Nicholas M. Orlans Peter T. Higgins, "Biometrics", McGraw-Hill/Osborne, ISBN: 0-07-223030-4, 2003

[2] A. K. Jain, P. Flynn, A. A Ross, " Handbook of Biometrics", Springer, USA, ISBN-13: 978-0-387-71040-2, pp.:1-23, 2007

[3] http://en.wikipedia.org/wiki/Biometrics , Accessed on 07.07.2010 , 10:42AM

[4] O.Rosso, S. Blanco, J. Yordanova,V. Kolev, A. Figliola, M. Schurmann, E. Bas,"Wavelet entropy: a new tool for analysis of short duration brain electrical signals",Journal of Neuroscience Methods 105 (2001) 65–75,Elsevier Science

[5] Bergadano, Gunetti and Picardi, "User Authentication through keystroke dynamics",ACM Trans. Information and System Security, Nov 2002

[6] F.Monrose, A.D. Rubin,"Keystroke dynamics as a biometric for authentication", Future Generation Computer Systems 16 (2000) 351–359, Elsevier,2000

[7] Fabian M, Rubin M., "Keystroke Dynamics as a Biometric for Authentication", Elsevier Publication, March 1999