Expert Systems with Applications 36 (2009) 10649–10656

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Improving authentication accuracy using artificial rhythms and cues for keystroke dynamics-based authentication

Seong-seob Hwang [a], Hyoung-joo Lee [a,b,*], Sungzoon Cho [a]

[a] Department of Industrial Engineering, Seoul National University, San 56-1, Shillim-dong, Daehakdong, Kwanak-gu, Seoul 151-744, Republic of Korea
[b] Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK

ARTICLE INFO

Keywords:
Keystroke dynamics
Biometrics
User authentication
Data quality
Artificial rhythms
Tempo cues
Novelty detection

ABSTRACT

Keystroke dynamics-based authentication (KDA) is to verify a user's identity using not only the password but also keystroke dynamics. With a small number of patterns available, data quality is of great importance in KDA applications. Recently, the authors have proposed to employ artificial rhythms and tempo cues to improve data quality: consistency and uniqueness of typing patterns. This paper examines whether improvement in uniqueness and consistency translates into improvement in authentication performance in real-world applications. In particular, we build various novelty detectors using typing patterns based on various strategies in which artificial rhythms and/or tempo cues are implemented. We show that artificial rhythms and tempo cues improve authentication accuracies and that they can be applicable in practical authentication systems.

1. Introduction

The password-based authentication is the most commonly used in identity verification. However, it becomes vulnerable when a password is stolen. Keystroke dynamics-based authentication (KDA) was proposed to provide additional security (Gaines, Lisowski, Press, & Shapiro, 1980). KDA was motivated by the observation that a user's keystroke patterns are repeatable and distinct from those of other users (Umphress & Williams, 1985). Its potential applications include internet banking, ATM machines, digital doorlocks, and cellular phones, which require high security. It is possible to complement the password-based authentication using other biometric attributes such as fingerprint, iris, and voice (Jain, Bolle, & Pankanti, 1999; Polemi, 1997). However, these methods require very expensive devices (Monrose & Rubin, 2000). In addition, users may be reluctant to provide those biometric data. On the other hand, KDA requires no additional device and involves little user discomfort (de Ru & Eloff, 1997; Monrose, Reiter, & Wetzel, 2002; Monrose & Rubin, 2000). For recent reviews on KDA, see Monrose and Rubin (2000), Peacock, Ke, and Wilkerson (2004).

There are three steps involved in KDA as illustrated in Fig. 1. First, a user enrolls his/her keystroke patterns. A keystroke pattern is defined as depicted in Fig. 2. A password of m characters is transformed into a $(2m + 1)$-dimensional timing vector. A "duration" denotes a time period during which a key is pressed while an "interval" is a time period between releasing a key and stroking the next key. Second, a classifier is built using the keystroke patterns. Third, when a new keystroke pattern is presented it is either accepted or rejected by the classifier.

One of the most obvious difficulties in KDA from a pattern recognition point of view is that impostor patterns are not available when building a classifier. Thus it is not possible to train a binary classifier. This limitation can be overcome by the novelty detection framework (Cho, Han, Han, & Kim, 2000; Lee & Cho, 2007; Yu & Cho, 2004). In novelty detection, the valid user's patterns are designated as normal and all other possible individuals' patterns as novel. A novelty detector learns the characteristics of normal patterns during training and detects novel patterns that are different from the normal ones during test. In a geometric sense, a novelty detector defines a closed boundary around the normal patterns in the input space (Japkowicz, 2001; Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001).

Another difficulty in KDA stems from the fact that in practice, the number of the valid user's patterns is limited. When a large number of typing patterns are available, complex algorithms such as neural network (Bishop, 1995) and support vector machines (SVMs) (Vapnik, 1998) can be built. When only a small number of typing patterns are available, on the other hand, simple algorithms such as k-nearest neighbor (Knorr, Ng, & Tucakov, 2000) and K-means (Lee & Cho, 2007) have to be adopted. However, a small number of patterns usually result in low accuracies. It is not realistic to ask a user to provide hundreds of patterns in

* Corresponding author. Address: Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK. Tel.: +44 1865 283153; fax: +44 1865 273908.
E-mail addresses: hss9414@snu.ac.kr (S.-s. Hwang), hjlee@robots.ox.ac.uk (H.-j. Lee), zoon@snu.ac.kr (S. Cho).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.02.075

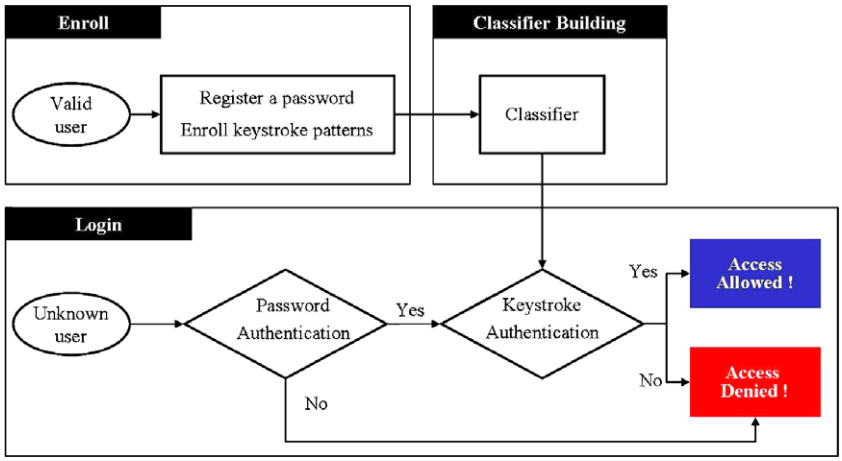

Fig. 1. Three steps of KDA framework: enrollment, classifier building, and login (user authentication).

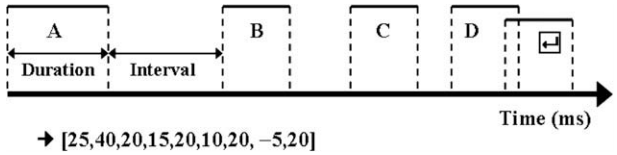

Fig. 2. A keystroke pattern is transformed into a timing vector when a user types a string 'ABCD'. The duration and interval times are measured by milliseconds.

keystroke enrollment. In order to address this problem, we have to improve the quality of patterns since improving data quality could be far more effective than finding a superior technique.

To make matters worse, users do change their passwords every once in a while and may adopt different passwords for different accounts. Therefore, it is not unusual that a password is newly adopted and/or relatively unfamiliar to the user. Unfamiliar passwords are usually translated into inconsistent keystroke patterns. In a preliminary experiment, equal error rates (EERs) of 25 users increased from 2.3% for familiar passwords to 11.7% for unfamiliar ones.

Recently, artificial rhythms and tempo cues were proposed to improve the quality of patterns: uniqueness and consistency in particular (Cho & Hwang, 2006). Uniqueness refers to how different the valid user's keystroke patterns are from those of potential impostors. Consistency is concerned with how similar the user's patterns in the authentication stage are to those enrolled in the enrollment stage. Kang, Park, Hwang, Lee, and Cho (2008) empirically showed that artificial rhythms increased uniqueness while cues increased consistency.

This paper examines whether improvement in uniqueness and consistency by implementing artificial rhythm and tempo cues translates into improvement in authentication performance in real-world off-line applications. First, artificial rhythms are shown to increase authentication accuracy. Second, we also show that artificial rhythms, if coupled with tempo cues, increase accuracy even more. Third, we show that artificial rhythms and cues are especially beneficial to users who are not good at typing.

The organization of this paper is as follows. The following section introduces strategies to improve data quality in KDA. Section 3 describes how data were collected for the experiments and various novelty detectors used in our experiments. In Section 4, various novelty detectors based on various strategies are compared in terms of accuracies. Finally, conclusions and future work are discussed in Section 5.

2. Strategies to improve the data quality in KDA

This section introduces data quality measures for typing patterns: uniqueness, consistency, and discriminability. Also discussed are strategies to improve the quality measures by employing artificial rhythms and tempo cues, and previous research on the conceptual effectiveness of the strategies.

2.1. Measures of data quality

Data quality in KDA can be measured in terms of uniqueness, consistency, and discriminability (Cho & Hwang, 2006). Uniqueness is concerned with how different a valid user's typing patterns used to build a classifier (we refer these typing patterns to enroll typing patterns) are from those of potential impostors' (we refer these typing patterns to impostor typing patterns). The more different enroll typing patterns are from impostor typing patterns, the easier a classifier can classify a valid user from impostors. For example, let "ABCD" be a password. Users usually type this password as they normally type those characters. However, if one types "abc" and pauses for, say, three beats before typing "d," this typing pattern becomes unique, because it has very different keystroke dynamics from potential impostors' typing patterns. Let $\{\vec{x}_i | i = 1,\ldots,N_x\}, \{\vec{y}_j | j = 1,\ldots,N_j\}$, and $\{\vec{z}_k | k = 1,\ldots,N_k\}$ denote enroll typing patterns, access typing patterns, and impostor typing patterns, respectively. Given the prototype pattern $\vec{m} = \sum \vec{x}_i / N_x$, uniqueness can be defined as

$$\text{Uniqueness} = \sum_{k=1}^{N_z} \frac{|\vec{z}_k - \vec{m}|}{N_z} - \sum_{i=1}^{N_x} \frac{|\vec{x}_i - \vec{m}|}{N_x}, \quad (1)$$

where $|\cdot|$ is a Euclidean distance. A high uniqueness will result in a low false acceptance.

Consistency is concerned with how similar a valid user's access typing patterns are to his enroll typing patterns. Since a classifier is built based on enroll typing patterns, even a valid user would be rejected if access typing patterns were not similar enough to enroll typing patterns. Consequently, improving consistency can also increase the performance of a classifier. We defined inconsistency rather than consistency for computational convenience as follows:

$$\text{Inconsistency} = \sum_{j=1}^{N_y} \frac{|\vec{y}_j - \vec{m}|}{N_y} - \sum_{i=1}^{N_x} \frac{|\vec{x}_i - \vec{m}|}{N_x}. \quad (2)$$

A low inconsistency will result in a low false rejection.