User Authentication through Keystroke Dynamics[1]

FRANCESCO BERGADANO, DANIELE GUNETTI, and CLAUDIA PICARDI
University of Torino

Unlike other access control systems based on biometric features, keystroke analysis has not led to techniques providing an acceptable level of accuracy. The reason is probably the intrinsic variability of typing dynamics, versus other—very stable—biometric characteristics, such as face or fingerprint patterns. In this paper we present an original measure for keystroke dynamics that limits the instability of this biometric feature. We have tested our approach on 154 individuals, achieving a False Alarm Rate of about 4% and an Impostor Pass Rate of less than 0.01%. This performance is reached using the same sampling text for all the individuals, allowing typing errors, without any specific tailoring of the authentication system with respect to the available set of typing samples and users, and collecting the samples over a 28.8-Kbaud remote modem connection.

Categories and Subject Descriptors: D.4.6 [**Operating Systems**]: Security and Protection—*access controls, authentication*

General Terms: Experimentation, Security

Additional Key Words and Phrases: Biometric techniques, keystroke analysis

1. INTRODUCTION

Biometric features (and techniques [Ashbourn 2000a]) are conveniently divided into two main categories. The *physiological* features include face, eye (normally, retinal or iris patterns), fingerprints, palm topology, hand geometry, wrist veins and thermal images. The *behavioral* features include voiceprints, handwritten signatures and keystroke dynamics [Polemi 2000].

In general, physiological features have been more successful than behavioral features to implement authentication systems based on one of such characteristics. This is not difficult to understand: physiological features essentially do not vary along time, whereas behavioral features such as signature and keystroke dynamics may change greatly even between two consecutive samplings (voice is a behavioral feature, but it is probably more stable than others, especially if samples are provided with the same uttered text).

[1] Patent pending.

Authors' address: Dipartimento di Informatica, University of Torino, corso Svizzera 185, 10149 Torino, Italy; email: {bergadan,gunetti,picardi}@di.unito.it.

On the other hand, many biometric techniques—and in particular the physiological ones—require specific tools (such as special videocameras) to sample the corresponding biometric feature. In the case of access control to computers, the need of an additional sampling tool limits the possibility to apply the technique: costs increase, and it is not obvious how to implement it in the case of remote connections.

Unlike other biometric methods, keystroke analysis can be done without the aid of special tools, just the keyboard of the computer where the biometric analysis has to be performed.[2] Nevertheless, user authentication through keystroke characteristics remains a difficult task. The reason is probably simple: physiological features such as face, retinal and fingerprint patterns are strongly stable over time, unlike behavioral features such as writing and typing dynamics. For these two features, a significant variability takes place normally, even without any evident change in the psychological and physiological state of the individual under observation. Especially for keystroke dynamics, variability between two immediately consecutive samplings occurs even if the subject providing the samples strives to maintain a uniform way of typing. Of course, nothing similar occurs with physiological characteristics. To some extent, even with voiceprints and handwriting it is possible to provide a certain level of uniformity. On the other hand, when typing on a keyboard it is pretty difficult to have some control on the number of milliseconds we hold down a key.

Nonetheless, much research on keystroke analysis has been done in the last years (e.g., Joyce and Gupta [1990], Bleha et al. [1990], Leggett et al. [1991], Brown and Rogers [1993], Obaidat and Sadoun [1997b], and Monrose and Rubin [1997]) which led also to some U.S. patents (such as Garcia [1986], Young and Hammon [1989], and Brown and Rogers [1996]). Also, more recently there has been an increased interest in the use of keystroke biometrics in conjunction with more classical access control techniques: keystroke analysis is used to harden passwords (such as in Reiter et al. [1999] and in *Biopassword*©, a software tool for Windows NT commercialized by Net Nanny Sw. Inc.[3]). It is clear that, in this case, part of the security is still enforced by the password.

In this article, we address the problem of user authentication via keystroke dynamics presenting a new keystroke analysis technique that should help to solve two of the problems related to keystroke analysis: the mentioned intrinsic variability of typing and the possibility of typing errors. We have extensively tested our approach on 154 individuals, achieving the best results among those found in the literature: a False Alarm Rate of about 4% and an Impostor Pass Rate of less than 0.01% (less than one successful attack out of 10,000 attempts). This performance is reached using the same sampling text of 683 characters for all the individuals, requiring a small number of samples to every user, allowing typing errors, and without any specific tailoring of the authentication system

[2] It is true that special keyboards may allow to measure additional features such as the acceleration or the energy impressed to the keystrokes, but such keyboards are normally not available, and expensive. As a consequence, it is important to be able to use only those features that can be sampled on normal keyboards, without the aid of specific devices.

[3] www.biopassword.com.

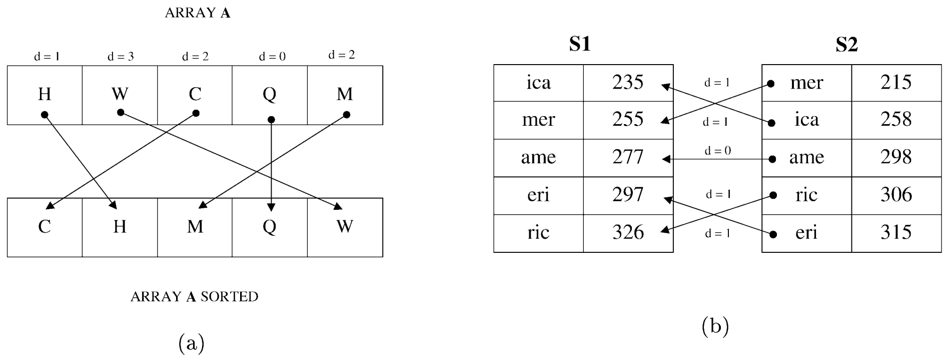


Fig. 1. (a) An array and its sorted counterpart. (b) Computation of the distance of two typing samples of the same text.

with respect to the available set of samples. We have performed the keystroke sampling through a 28.8-Kbaud phone line. In this way, we want to prove that keystroke analysis can be practically and successfully adopted for applications that require a reasonable level of accuracy even over remote connections.

We describe the technique and the experiments we have done in the next sections, and then we discuss our approach with respect to the length of the sample text, some experimental properties of the method, and its scalability. Finally, we compare our results with other approaches found in the literature, we illustrate some possible applications of our method, and conclude.

2. DEGREE OF DISORDER OF AN ARRAY

Given a array V of N elements, a simple measure of the *degree of disorder* (or, simply, the *disorder*) of V with respect to its ordered counterpart V′ can be computed as the sum of the distances between the position of each element in V and the position of the same element in V′. As an example, the (degree of) disorder of array A in Figure 1(a) is: $(1 + 3 + 2 + 0 + 2) = 8$.

Hence, a sorted array V′ has a degree of disorder equal to 0, while we have the maximum disorder for an array V″ when its elements are in reverse order. It is easy to see that such value of disorder is given by:

$$\frac{|V''|^2}{2}\ (\text{if } |V''| \text{ is even});\ \frac{(|V''|^2 - 1)}{2}\ (\text{if } |V''| \text{ is odd})$$

Given an array of N elements, it is convenient to normalize its degree of disorder by dividing it by the value of the maximum disorder of an array of N elements. In this way it is possible to compare the disorder of arrays of different size. After this normalization, it is clear that, for any array V, its degree of disorder falls between 0 (if V is ordered) and 1 (if V is in reverse order). For the array A of Figure 1(a), we have: $(1 + 3 + 2 + 0 + 2)/((5^2 - 1)/2) = 8/12 = 0.66666$.

The distribution of all the possible arrays of N different elements with respect to their (normalized) disorder is not uniform. In particular, disorder takes a precise form, as depicted in Figures 2(a) and 2(b). Note that the vertical axes

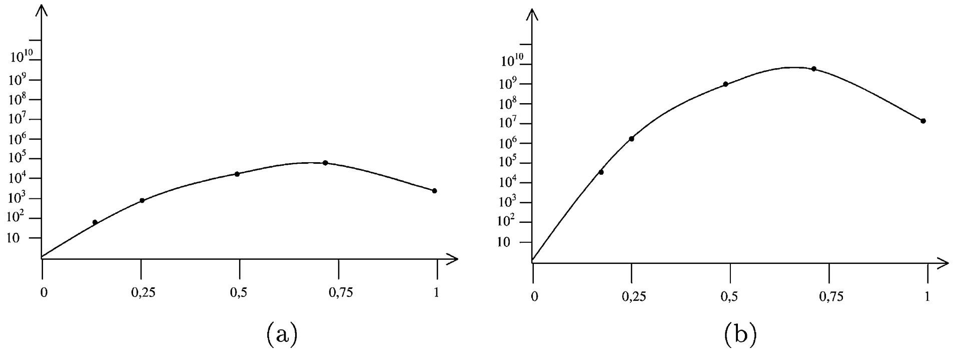


Fig. 2. (a) Distributions of arrays of 9 elements. (b) Distributions of arrays of 13 elements.

have a logarithmic scale. Arrays accumulate mainly in the interval [0.5–1], and especially in the interval [0.5–0.75]. Moreover, the ratio between the number of arrays with disorder higher than 0.5 and the number of arrays with disorder below 0.5 increases with n. For instance, given 9 different elements, and hence 9! different arrays that can be built from them, the ratio between the number of arrays with disorder larger than 0.5 and the number of arrays with disorder smaller than 0.5 is 4.478. This ratio increases to 8.075 for arrays of 13 elements. We generated 10 million different arrays of 100 elements, randomly sorted. Only 573 had a disorder in the interval [0.44–0.5]. The remaining had a disorder in the interval [0.5–0.9]. This property is important, since it can be used to compare two typing samples in order to decide if they have been provided by the same user or by different individuals.

3. DISTANCE OF TWO TYPING SAMPLES

Given two typing samples of the same text we want to quantify their "similarity" or "difference." In other words, we want to define a measure of the *distance* of the two samples. Experiments found in the literature (e.g., in Umphress and Williams [1985], Joyce and Gupta [1990], Brown and Rogers [1993], and Obaidat and Sadoun [1997]) normally use some combination of two basic measures: (1) the *duration* of a key (how long a key is held down) and (2) the *latency* between two consecutively typed keys (the elapsed time between the release of the first key and the depression of the second). Two keys typed one after the other are called a *digraph*. Similarly, three consecutively typed keys are called a *trigraph*. In our experiments, we only took into consideration a pretty rough measure: the elapsed time between the depression of the first and of the third key of a trigraph. We call such measure the *duration* of the trigraph.

As an example, suppose that a user is asked to type the text: *america*. The outcome of the sampling, when using trigraphs, could be the following (the semicolon is only used to separate the trigraphs, and it is not part of what is typed):

S1: ame 277; mer 255; eri 297; ric 326; ica 235

Next to each trigraph is its duration in milliseconds.[4] A given sample can be sorted with respect to the duration of its trigraphs. As a consequence, **S1** becomes:

S1: ica 235; mer 255; ame 277; eri 297; ric 326

From now on, when speaking of a typing sample we mean an array of trigraphs sorted with respect to their duration. Now, suppose a second sample **S2** of the same text is provided:

S2: mer 215; ica 258; ame 298; ric 306; eri 315

We may consider **S1** as the referring sorted array, and we may compute the *distance* of **S2** with respect to **S1** (in short: d(**S1**,**S2**)), as the degree of disorder of **S2** with respect to **S1**. In other words, the distance of **S2** from **S1** is the sum of the distances of each trigraph of **S2** with respect to the position of the same trigraph in **S1**. It is clear that d(**S1**,**S2**) = d(**S2**,**S1**). Figure 1(b) shows graphically an example of how d(**S1**,**S2**) is computed. The absolute (degree of) disorder of **S2** with respect to **S1** is $1 + 1 + 0 + 1 + 1 = 4$, and the maximum disorder of an array of 5 elements is 12. Hence, the *normalized distance* of **S2** from **S1** is:

$$\mathrm{d}(\mathbf{S1},\mathbf{S2}) = \mathrm{d}(\mathbf{S2},\mathbf{S1}) = \frac{(1+1+0+1+1)}{12} = 0.33333$$

Note that we do not take into consideration the duration of the trigraphs of **S1** and **S2**, but just the relative position of the trigraphs after the two samples have been sorted. As a consequence, for a sample like the following:

S3: ica 125; mer 338; ame 391; eri 402; ric 415

it would be d(**S1**,**S3**) = 0, d(**S2**,**S3**) = 0.33333.

For longer texts, and for a timing resolution larger than one millisecond (which is the case for our experiments, as described in the next section), it may happen that different trigraphs have the same duration. In this case, the trigraphs are sorted in alphabetical order.

4. EXPERIMENTAL SETTING

To test the ability of the distance measure described in the previous section to discriminate users through their typing dynamics, we asked 44 persons in our Department to type five times a fixed text of 683 characters, for a total of 220 samples.[5] In this section, we give a detailed description of the way we gathered the samples.

[4] In general, a given trigraph may be typed more than once. In that case the mean of the duration of the trigraphs is used.

[5] Besides, as we describe in Section 6.1, another 110 persons were asked to provide one sample of the same text.

4.1 The Sample Text

The text used to produce the samples was taken from the beginning of one of the most famous Italian novels, “I Promessi Sposi” (“The Promised Newlyweds”), plus a short text in English. However, with respect to the original text, we avoided capital letters (hence, the shift key had not to be used), we changed semicolons to commas, and we turned every stressed letter to the corresponding plain letter followed by an apostrophe (this last change is rather common for Italians using computer keyboards, even when using Italian keyboards). We also asked the volunteers to start new lines (i.e., typing a carriage return) according to the text they had to type in. As a consequence, each sample was produced using only the twenty-six lower-case letters, plus the space, the full stop, the comma, the apostrophe and the carriage return keys.[6]

4.2 The Volunteers

All the people participating in the experiments were native speakers of Italian and well familiar with the language of the text they were typing, which is written in plain Italian. They were also familiar with English, and used to typing English words and sentences.

The volunteers had, in general, heterogeneous typing skills, varying from professional secretaries to programmers using a computer every day, from computer science students to researchers and professors. However, all of them were experienced in typing on normal computer keyboards, and no one was so unskilled to have to look for characters on the keyboard. None of the volunteers was hired, or in any way paid for his/her typing speed: we asked the people just to type in the sample text as they would have done as a part of their normal job.

4.3 Gathering of the Samples

The samples were collected on the basis of the availability and willingness of people over a period of one month. No one provided two samples in the same day and, normally, a few days passed between two samples given by the same user. No volunteer was trained to type the sample text before the gathering of his/her five samples. All the samples where provided on the same keyboard of the same notebook, in the same office and with the same artificial light conditions. Each volunteer was left free to adjust the position of the notebook on the desk, position of the screen and chair height as preferred. The keyboard used in the experiments had the characters placed in the usual positions, and

[6] For the sake of completeness, here is the exact reproduction of the sample text:
quel ramo del lago di como, che volge a mezzogiorno, tra due catene non interrotte di monti,
tutto a seni e a golfi, a seconda dello sporgere e del rientrare di quelli, vien, quasi a un tratto,
a restringersi, e a prender corso e figura di fiume, tra un promontorio a destra,
e un’ampia costiera dall’altra parte.
e il ponte, che ivi congiunge le due rive, par che renda ancor piu’ sensibile all’occhio
questa trasformazione, e segni il punto in cui il lago cessa, e l’adda ricomincia,
per ripigliar poi nome di lago dove le rive, allontanandosi di nuovo,
lascian l’acqua distendersi e rallentarsi in nuovi golfi e in nuovi seni.

the quick brown fox jumps over the lazy dog.

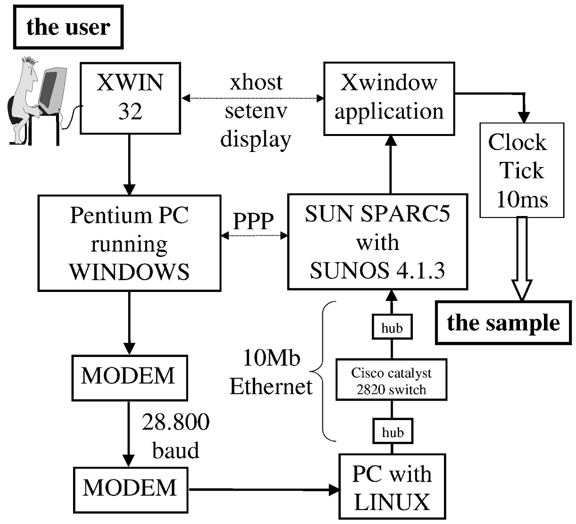


Fig. 3. Hardware setting for the experiments.

with a key size similar to those of normal desktop keyboards. However, that specific keyboard had never been used before by any of the volunteers. We are well aware that the gathering of all samples on the same keyboard is a rather artificial situation, since, in real applications, users (and impostors) will very likely use different computers, especially in case of remote connections to servers from different workstations. It is, however, unclear if and how this choice may have affected the outcomes of our experiments.

When an individual was providing a sample, the sample text to be reproduced was displayed on the top of the notebook screen, with the typed text appearing just below the referring text. Echoing was handled remotely by the machine collecting the timing data (see below), but perceived as instantaneous by the volunteers.

4.4 Hardware Setting

The machinery used to gather the samples was set as as depicted in Figure 3.

A Sun Workstation with SunOs 4.1.3 and X-window as the graphical interface was used to measure the trigraph durations. Sampling was made by an X-window application able to detect the depression of keyboard keys, and a system clock with a resolution of 10 milliseconds was used to measure the elapsed time between two consecutive interrupts generated by key depressions. All the participants to the experiments provided their samples on the keyboard of a Pentium notebook running Windows and an Xwin32 interface. The notebook communicated via a Point-to-Point Protocol connection to the Sparcstation through a chain including a 28.8-Kbaud modem connection, a PC running Linux and an Ethernet local network at 10 megabits. Information flowing between the Sparcstation and the Linux PC had to pass through a Cisco Catalyst 2820 Switch and two Hubs.

We chose this setting in order to test the reliability of our authentication method even when the keystroke sampling is performed over remote, relatively slow, connections.

We want to note that our experiments may have been affected by the traffic present on the local network when gathering a sample. All the samples were collected during working hours, when the departmental network is often heavily loaded. However, a few samples may have been provided when the local network was particularly busy, or conversely, pretty unloaded. Timing measurements, and hence outcomes, may have been affected to some extent by such transient situations.

4.5 Typing Errors

The distance between two samples is computed on the basis of the relative positions of the trigraphs the samples are made of. The relative position of a trigraph in a sample depends on the duration of that trigraph. As a consequence, when comparing two samples, they must contain the same trigraphs. However, this does not mean that the two samples must be produced by typing exactly the same text. Simply, before the samples are compared to compute their distance, they are filtered in order to keep only the shared trigraphs. Of course, the larger the number of common trigraphs, the more meaningful the value of their distance. If the same text is used for all the typing samples, the only trigraphs not shared by two samples are those due to typing errors. Such trigraphs are filtered away before comparing the two samples. If the number of trigraphs kept in the two samples is large enough, the computation can still take place.

In our experiments, each user was always left free to make typing errors, and to decide whether to correct them or not. Also, the user was free to stop typing as he/she liked (to reread what was written up to that point, to correct something, or just to take a break). No sample was thrown away because of typing errors in it. Of course, this had consequences on the number of trigraphs actually involved in the comparison of two samples: though the text used in the experiments is made of about 350 different trigraphs, the number of trigraphs shared by two samples was 272 on the average. In the whole set of samples used in our experiments, there is virtually no one pair of samples containing the same set of trigraphs.

It must be noted that most of the experiments found in the literature reject any sample containing typing errors (e.g., Bleha et al. [1990], Brown and Rogers [1993], and Obaidat and Sadoun [1997b]). However, in Leggett and Williams [1988], samples are kept even if they contain typing errors, while no information is available for the experiment described in Joyce and Gupta [1990].

5. USER CLASSIFICATION

Suppose we are given a set of users and a set of typing samples of the same text from those users. Given a new sample from one of the users, we want to determine who typed it. On the average, we may expect the distance between two samples of the same user to be smaller than the distance between two samples

Table I. Experimental Results in User Classification

N. of samples in the model of each user	4 samples	3 samples	2 samples	1 sample
Tot. N. of classifications attempted	220	880	1320	880
Tot. N. of errors	0	2	10	27
% of correct classification	100%	99.7727%	99.2424%	96.9318%

of different users. In a "perfect world," it may be the case that d(**S1**,**S2**) = 0 for two different samples of the same user, whereas in reality it might also occasionally happen that d(**S1**,**S2**) > d(**X1**,**Y2**) if **S1** and **S2** belong to the same user while **X1** and **Y2** come from different users. This happens because the way an individual types on the keyboard may vary according to different psychological and physical conditions, such as stress and tiredness. This is more true if we allow users to make typing errors.

As a consequence, the classification can be more accurate if more typing samples of each user are available to classify a new incoming sample. For instance, suppose three users A, B, and C provide, respectively, 4, 3, and 5 typing samples of the same text, and let X be a new sample provided by one of the three users. The samples of each user constitute what can be called a *model* (or *profile*) of that user: a model of the way he/she types on the keyboard. We compute the *mean distance* (*md* for short) of sample X from the model of each user as the mean of the distances of X from each sample of each user:

$$\begin{aligned}\mathrm{md(A,X)} &= \mathrm{(d(A1,X) + d(A2,X) + d(A3,X) + d(A4,X))/4}\\ \mathrm{md(B,X)} &= \mathrm{(d(B1,X) + d(B2,X) + d(B3,X)))/3}\\ \mathrm{md(C,X)} &= \mathrm{(d(C1,X) + d(C2,X) + d(C3,X) + d(C4,X) + d(C5,X))/5}\end{aligned}$$

and classify X as belonging to the user with the smallest mean distance.

5.1 Experimental Results in User Classification

The 220 samples collected were used to test the simple classification procedure described above. If X is one of the 220 samples, we compute the mean distance of X from the models of the 44 users, and classify X as belonging to the user whose mean distance of X from his/her model is the smallest.

The results of this experiments are shown in Table I. By using four samples for each user (the remaining one being the one to be classified), we get a 100% of correct classification. It is to be expected that a smaller number of samples in the model of each user should lower the performance of the classification procedure. To check this, we repeated the same classification experiment using only 3, 2 and 1 sample as a model of every user.[7] It is easy to see that still the classification procedure performs very well. Even when using just one typing sample of a user, we get a percentage of correct classification of almost 97%.

[7] By using three samples (out of five available) for a user's model, it is possible to build ten different models for each user, and every model can be tested using the remaining two samples. As a consequence, for the 44 users we have a total of 880 attempted classifications. A similar situation holds when using two and one sample in the model of a user.

6. USER AUTHENTICATION

An Access Control system based on any biometric measure has to perform a task that is much more difficult than a classification task such as the one described in the previous section. There, a person who provides a new sample has to be correctly identified among a finite set of known users. By contrast, an Access Control system must be able to deal with incoming samples that may belong to one of the legal users or to impostors, who have completely unknown biometric features. Of course, for practical applications we expect the FAR of the system to be small and, above all, the IPR to be negligible.[8] To deal with unknown impostors, the system must be able to accept/reject a new sample X, provided by someone who claims to be the legal user U, not only if X is closer to U's model than to any other model: after all, this could happen by chance. A sample X must be *sufficiently close* to U's typing model in order to be classified as a sample of U.

In this section, we describe the experimental results achieved in user authentication through the use of the biometric measure and the typing samples described in the previous sections. First, we need to describe the experimental setting.

6.1 Experimental Setting in User Authentication

The 44 persons who provided five samples were used as legal users of a hypothetical system. Still in the same experimental setting described in Section 4, we asked another 110 persons to provide only one sample, to be used as an intrusion attempt.

For each sample S of each legal user U, we do the following:

—The four samples of U not including S are used as the model of U known to the system.

—S is used as a new sample provided by the user U who is trying to connect to the system (i.e., S in unknown to the system, and should hopefully be recognized as belonging to U on the basis of the other samples of U).

—User U is attacked 110 times by 110 different persons who pretend to be U, using the sample provided by those persons. Moreover, user U is attacked 215 times using all the samples of the other legal users of the system.[9] Therefore, we have one legal attempt of user U to enter the system with sample S, and a total of $110 + 215 = 325$ attacks brought by $110 + 43 = 153$ potential intruders who are trying to enter the system pretending to be U.

[8] We recall that, when testing the performance of an access control system based on biometric features, the *False Alarm rate* (*FAR*) is the percentage of connections attempted by legal users of the system erroneously rejected. The *Impostor Pass Rate* (*IPR*) is the percentage of successful attacks brought to the system by impostors pretending to be one of the legal users. Of course, in both cases, the lower the percentage, the better.

[9] When a sample S1 of a legal user U1 is used to attack another legal user U, all the samples of U1 are temporarily removed from the system before running the authentication procedure, otherwise S1 would be very likely recognized as belonging to U1, and the attack would almost certainly fail. In other words, when a sample of U1 is used to attack U, U1 becomes completely unknown to the system.

Hence, on the whole, the system is tested 220 times (by 44 users) with legal connections attempts, and 71500 times (by 154 potential intruders) with intrusion attempts.[10]

6.2 Experimental Results in User Authentication with the Classification Procedure

Since the classification procedure of Section 5 performs very well, one may just guess that such a procedure could as well be used to classify a new sample and allow the user who provided it to enter the computer, or to deny access.

Unfortunately, such control can be easily fooled. An impostor may just pretend to be one of the legal users, chosen randomly, and provide his sample. If there are N legal users on the system, the impostor has one chance out of N of being erroneously recognized as the legal chosen user. This happens if, by accident, the impostor's sample is closer to the legal user's model than to any other system's user. In other words, even if the classification procedure performs perfectly, showing a 0% FAR, it could at best have (100/N)% IPR, if there are N legal users known to the system.

The column of Table II with $k = 1$ (the meaning of k will be explained in the next section) illustrates perfectly such situation, where the classification procedure is used to grant or deny access. Every new sample belonging to a legal user is correctly recognized, but 1650 attacks, out of 71500 are successful.

As a consequence, we achieve an FAR of 0%, but an IPR of 2.3077%. That is, roughly one attack out of 44 succeeds (In fact, note that 100/44 = 2.2727). Actually, with an FAR = 0%, an IPR of 2.3% is not all that bad, but much better results can be achieved through a simple observation: an new sample X is likely to belong to user A only if X has a distance from the samples in A's model similar to the distance between any two such samples. This is formalized in the following section.

6.3 Use of Thresholds in the Classification Procedure

The classification procedure classifies a new sample as belonging to the user whose mean distance from the known samples of the legal users (i.e., their models) is the smallest. However, this may produce some counterintuitive effects. Suppose for instance that we have a sample X of someone who maintains to be user A, and samples A1, A2, A3, B1, B2, B3, C1, C2, C3 belonging to users A, B and C respectively. Suppose we compute:

$$\mathrm{md(A,X)} = (\mathrm{d(A1,X)} + \mathrm{d(A2,X)} + \mathrm{d(A3,X)}))/3 = 0.419025$$
$$\mathrm{md(B,X)} = (\mathrm{d(B1,X)} + \mathrm{d(B2,X)} + \mathrm{d(B3,X)}))/3 = 0.420123$$
$$\mathrm{md(C,X)} = (\mathrm{d(C1,X)} + \mathrm{d(C2,X)} + \mathrm{d(C3,X)}))/3 = 0.423223$$

[10] Someone could observe that the five attacks brought by the five samples of each legal user when used as an impostor may partially undermine the validity of the experimental outcomes. In fact, it is true that if one of the samples of an attacker fails to pass the test as belonging to someone else, even the other samples of the same attacker are likely to fail. However, also the converse applies: if one of the samples of an attacker passes the test as belonging to someone else, even the other samples of the same attacker are likely to pass. Similar approaches are adopted in Joyce and Gupta [1990] and Obaidat and Sadoun [1997].

Table II. Results in User Authentication Using the Classification Procedure with Different Values of k

Value of k	$k = 1$	$k = 0.66$	$k = 0.5$	$k = 0.33$	$k = 0.3$
N. of successful attacks out of 71500 attempts	1650	98	30	2	0
N. of failed legal connections out of 220 attempts	0	0	4	13	16
Impostor Pass Rate	2.3077%	0.1371%	0.042%	0.0028%	0%
False Alarm Rate	0%	0%	1.8182%	5.9091%	7.2727%

and hence X is classified as belonging to user A. However, suppose that the mean of the distances of the samples forming the model of A (denoted by m(A)) is:

$$d(A1,A2) = 0.312378;\ d(A1,A3) = 0.304381;\ d(A2,A3) = 0.326024;$$
$$m(A) = (\ 0.312378 + 0.304381 + 0.326024\ )/3 = 0.314261$$

Then, we expect another sample of A to have a mean distance from the model of A similar to m(A), which is not the case for X in the example above.

Hence, we may refine the classification procedure as follows: a new sample X, claimed to be from user A, is classified as belonging to A if and only if: (1) md(A,X) is the smallest and (2) md(A,X) is closer to m(A) than to any other md(B,X) computed by the system. That is, md(A,X) must be smaller than $m(A) + |0.5(md(B,X) - m(A))|$ for any user B.

More generally, the classification rule may be rewritten as:

$$md(A,X) < m(A) + |k(md(B,X) - m(A))|$$

where B is another user of the system such that md(B,X) is the second closest value to m(A) after md(A,X), and k is a numeric constant. As just observed, if $k = 0.5$, we simply ask md(A,X) to be closer to m(A) than to any other md(B,X). A value for k such as $k = 0.66$ would be a slightly weaker requirement, and $k = 0.33$ would ask for a stronger evidence of X being a sample of A (in fact, with $k = 0.33$ we ask the distance md(A,X) to be twice as close to m(A) as to md(B,X)). If $k = 1$ we have the plain classification procedure of Section 5.

Of course, such a refinement may cut away some samples from legal users, but it is also likely to help ruling off impostors' attacks. Table II shows the IPR and FAR obtained in user authentication using the classification procedure with different standard values for k.

From the results of Table II, it is easy to see that, essentially, good results can be obtained choosing any value between 0.33 and 0.66 for k. Of course, small values of k provide better security, but also higher chances for samples from legal users to be rejected. Probably, the best trade-off between IPR and FAR is achieved with $k = 0.5$. Such a value for k is also a very reasonable and easy to understand rule of thumb: a sample provided under the identity of user A is accepted if it is closer to the model of A than to the model of any other user in the system.

We have also computed the largest value for k that still gives an IPR $= 0\%$ (last column of Table II). Note that the corresponding FAR still remains quite acceptable. Similarly, the largest value for k that allows for a FAR $= 0\%$ is also able to provide a very low IPR.

6.4 Adding Additional Filters

As we just saw, the use of thresholds improves the ability of the classification procedure to recognize legal users and reject impostors, however, it may have some counterintuitive behavior. Consider again the example of Section 6.3, but now with the following values:

$$\mathrm{md(A,X)} = (\mathrm{d(A1,X)} + \mathrm{d(A2,X)} + \mathrm{d(A3,X)}\ ))/3 = 0.307025$$

$$\mathrm{md(B,X)} = (\mathrm{d(B1,X)} + \mathrm{d(B2,X)} + \mathrm{d(B3,X)}\ ))/3 = 0.420123$$

$$\mathrm{md(C,X)} = (\mathrm{d(C1,X)} + \mathrm{d(C2,X)} + \mathrm{d(C3,X)}\ ))/3 = 0.423223$$

$$\mathrm{d(A1,A2)} = 0.212378;\ \mathrm{d(A1,A3)} = 0.204381;\ \mathrm{d(A2,A3)} = 0.226024;$$

$$\mathrm{m(A)} = (\ 0.212378 + 0.204381 + 0.226024)/3 = 0.214261$$

(md(A,X) is much smaller now, but also the mean distance among the samples in A's model is smaller). Suppose also that that the classification procedure is used with $k = 0.5$, so that we have:

$$0.307192 < 0.212378 + |0.5^*(0.420123 - 0.307192)|$$

As a consequence, X is attributed to user A. However, if we look at the average distance of two samples of A, we see that X still does not behave as expected, if compared to the other samples from A. The mean distances of A1, A2 and A3 between each other is 0.214261, whereas the mean distance between X and A1, A2 and A3 is 0.307192, which is much larger. As a consequence, in this case, it would be reasonable to reject X. In other words, X should be sufficiently close to A's model *not only* with respect to any other user B: it must be sufficiently close to the samples in the model of A.

We have tested different ways to express this concept of *closeness* of a sample to a given model. As already noted, it would be nice to be able to filter out as many impostors' samples as possible, at the same time avoiding to reject samples from legal users. One *filter* that we have experimented with that performs well is the following:

Let A1, A2, A3, A4 be four samples provided by user A, and let mAxyz be the mean distance of samples x,y and z of A. As an example, mA123 is:

$$\mathrm{mA123} = (\mathrm{d(A1,A2)} + \mathrm{d(A1,A3)} + \mathrm{d(A2,A3)}/3).$$

For each of the four samples, we may compute the mean distance of that sample with respect to the other samples of A:

$$\mathrm{dA1} = |(\mathrm{d(A1,A2)} + \mathrm{d(A1,A3)} + \mathrm{d(A1,A4)})/3 - \mathrm{mA234}|$$

$$\mathrm{dA2} = |(\mathrm{d(A2,A1)} + \mathrm{d(A2,A3)} + \mathrm{d(A2,A4)})/3 - \mathrm{mA134}|$$

$$\mathrm{dA3} = |(\mathrm{d(A3,A1)} + \mathrm{d(A3,A2)} + \mathrm{d(A3,A4)})/3 - \mathrm{mA124}|$$

$$\mathrm{dA4} = |(\mathrm{d(A4,A1)} + \mathrm{d(A4,A2)} + \mathrm{d(A4,A3)})/3 - \mathrm{mA123}|.$$

Table III. Results in User Authentication Using the Classification Procedure and an Additional Filter

value of k	$k = 0.5$	$k = 0.5$	$k = 0.5$	$k = 0.5$	no k	$k = 0.55$
value of a	$a = 1$	$a = 1$	$a = 1.5$	$a = 1.5$	$a = 1.5$	$a = 1.22$
value of b	$b = 1.5$	$b = 1.75$	$b = 0$	$b = 0.5$	$b = 0.5$	$b = 1.25$
Successful attacks (out of 71500)	3	5	4	7	1032	7
Failed legal connections (out of 220)	12	9	10	8	5	4
IPR	0.0042%	0.007%	0.0056%	0.0098%	1.4433%	0.0098%
FAR	5.4545%	4.0909%	4.5454%	3.6364%	2.2727%	1.8182%

Then, let:

$$\text{MAX\_d(A)} = \text{MAX(dA1, dA2, dA3, dA4)};$$
$$\text{sd\_d(A)} = \text{standard\_deviation(dA1, dA2, dA3, dA4)}.$$

These two values can be used to accept a sample X as a new sample of A only if:

$$\text{md(A,X)} < \text{m(A)} + a * \text{MAX\_d(A)} + b * \text{sd\_d(A)},$$

where a and b are two constants that should be chosen in order to have an acceptable balance between IPR and FAR. The rationale of the filter is that we use the available samples to compute the maximum distance between a given sample and the mean of the remaining ones (i.e., MAX_d(A)). Then, we expect a new sample of A to have at most a similar distance from the model of the user. The standard deviation sd_d(A) and the constants a and b can be used to vary the threshold of acceptance/rejection in order to have more or less stringent requirements.

Table III shows the results in user authentication when applying the filter just described *after* the classification procedure, with $k = 0.5$ and for some *reasonable* values of a and b. The last but one column in the table shows the results in user authentication using only the filter described in this section: the FAR slightly improves, but the IPR worsens very much.

From Table III, we see that good results in user authentication are obtained for any of the chosen values for a and b. Of course, larger values for these constants give a better FAR but a worse IPR. Nonetheless, any of the first four sets of values for k, a and b could be successfully used to implement a biometric Access Control system.

Using only three samples in the model of each user (instead of four as in Table III), the performance of the authentication method decreases, as expected. In particular, the FAR roughly doubles, whereas the IPR increases from two to five times with respect to the values of the columns of Table III. Clearly, the larger the number of samples in the model of each legal user, the better the performance of the authentication system.

The last column of Table III shows the values for k, a and b that give the best results for the samples we collected (if we consider the total number of errors made by the system to accept or reject samples, and we also want to keep the FAR reasonably small). Clearly, the same values are very unlikely to perform so well also for other set of samples. On the contrary, a "reasonable" tuning for k, a and b (like those chosen in the other columns of Table III) is unable to provide the best performances, but there are higher chances that it will perform quite well also for other sets of data.

It can also be shown that our authentication system performs perfectly (i.e., IPR = FAR = 0%), at least on the collected set of samples. This can be achieved by carefully tuning k, a and b for every different legal user of the system, on the basis of the available samples. Of course, the system would hardly show the same perfect behavior on a different set of samples.[11] Nonetheless, tailoring the thresholds of acceptance and rejection of new samples for every legal user would be the right choice for real applications, when many samples are supposed to be available in each user's model. In that case, specific constants are very likely to outperform generic thresholds like those used in our experiments.

7. DISCUSSION

Unlike most other approaches found in the literature, the distance measure we have used in our research completely overlooks any absolute value of the temporal features measured in the typing samples. In fact, these absolute values (such as keystroke duration and latency) may greatly vary with the psychological and physiological state of the person providing the sample, but it is reasonable to expect the changes being homogeneous, affecting all of the typing characteristics in a similar way.[12] As a consequence, a measure that only considers the relative values of the various typing features should be less affected by psychological and physiological changes, and should perform well to extract and taking into consideration only the intrinsic typing features of the users.

Note that we have performed the experiments in a very homogeneous environment for all of the users, since the same text was typed by all of the volunteers. Clearly, this makes the job of the authentication system much harder. For example, in the experiments found in Brown and Rogers [1993] and Obaidat and Sadoun [1976], described later, each user types in many times a different string that is already very familiar to him/her, whereas the impostors type the same string (presumably quite unfamiliar to the attackers) for a much smaller number of times. Of course, this setting makes it easier to spot different typers

[11] Note that we are not claiming that such specific values can be found for any given set of users, their samples, and the samples used to attack them. However, such values do exist for the set of samples we collected in our experiments.

[12] For example, one day, a user is particularly nervous, or just in a hurry, and types on the keyboard more quickly than usual. The absolute temporal values of his/her keystrokes will vary very much, but the relative values would probably remain more stable. That is, if a user types the trigraph *his* more quickly than the trigraph *her*, this is likely to remain unchanged even in different typing conditions.

of the same string. On the contrary, the text used in our experiments was the beginning of a novel, and the number of samples provided and the interval between two samples from the same user left very few chances for a volunteer to get used to typing that text.

Finally, the distance measure appears to behave well even in spite of the complex set of machinery and programs through which the samples have been collected. This ability, even in a pretty unfavorable environment, is important for possible real applications. In fact, the samples were gathered in a situation that resemble the case of, for example, a user at home who wants to connect to a specific computer of the local network of his office using a normal phone line, as many of our colleagues do in our department (though, of course, office environment conditions may be quite different from those found at home). Clearly, the meaningfulness of gathered timing information would hardly survive heavy packet routing over the Internet as, for example, in the case of very remote telnet connections.

In the next sections, we discuss some specific aspects of the experiments described in this article.

7.1 On the Length of the Sample Text

Regardless the specific technique used to authenticate users through keystroke dynamics, it is clear that the reliability of the adopted method is closely related to the length of the text used to produce the typing samples. In fact, there is little information provided by two consecutive keystrokes, and we need many of them to discriminate among individuals. This is particularly true for our approach, since the distance between two typing samples is computed on the basis of the relative position of the trigraphs in the samples. In Section 7.2, we discuss some properties of the distance measure adopted in our approach. Here, we show the outcomes of some of the experiments we have done to test the performance of our method with respect to the number of trigraphs involved in the authentication task.

From each available sample, we have produced a new sample taking only a half of the trigraphs. More precisely, a new sample is made of the first half of the trigraphs of each original sample after it has been sorted alphabetically.[13] Note that this is slightly different from producing a new sample considering only the trigraphs of the first half of the original sample text. Moreover, we have done the same by taking only a quarter of the trigraphs in each available sample. We have then repeated all the experiments presented in the previous sections using the new samples, and some of the outcomes are shown in Table IV: IPR and FAR reached in user authentication using the classification procedure with different thresholds values (as in Section 6.3). In particular, the first entry of the table shows the same values of Table II. As expected, the smaller the number of shared trigraphs, the lower the ability of the system to

[13] Hence, if the original sample contains, among the others, the trigraphs *aut* and *uto*, because the word *auto* was typed, only the trigraph *aut* is taken in the new sample.

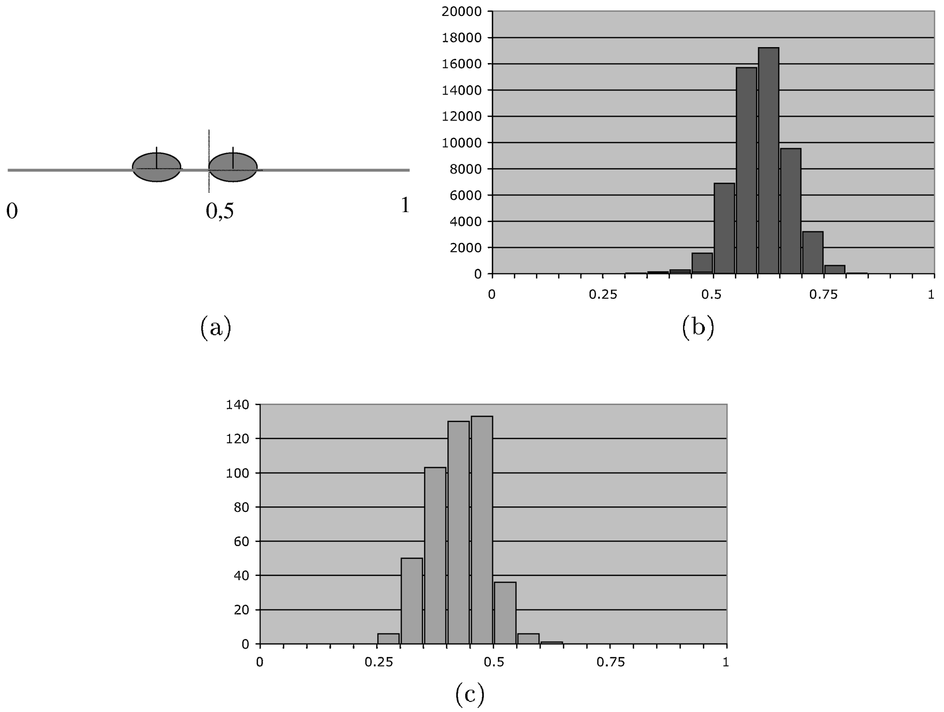
Table IV. Results in User Authentication using the Classification Procedure with Different Values of k and for Different Quantities of Shared Trigraphs

value of k		$k = 1$	$k = 0.66$	$k = 0.5$	$k = 0.33$
all the trighraphs	IPR	2.3077%	0.1371%	0.042%	0.0028%
(272 shared trigraphs on the avg.)	FAR	0%	0%	1.8182%	5.9091%
1/2 of the trighraphs	IPR	2.3077%	0.1413%	0.046%	0.0154%
(137 shared trigraphs on the avg.)	FAR	0.9091%	1.8182%	5.4545%	14.091%
1/4 of the trighraphs	IPR	2.3077%	0.3329%	0.1552%	0.08111%
(68 shared trigraphs on the avg.)	FAR	8.1818%	12.7273%	17.2727%	23.1818%

discriminate between impostors and legal users. However, consider, for example, the column for $k = 0.5$. Using only a half of the trigraphs of each sample (for an average 137 shared trigraphs in each comparison between any to samples), we still have less then one successful attack out of 2,000 temptatives, with a False Alarm Rate of about one error out of twenty legal connections. Also, in this case, the use of additional filters allows for some improvement, as seen in Section 6.4.

We have also repeated the experiments taking into consideration, from each original sample, only the trigraphs typed more than once. The idea is that the mean duration of a trigraph typed many times should be more representative of the way an individual types that trigraph, and thus may provide better performance. On the average, the number of trigraphs typed more than once and shared by two samples is 64,[14] but, quite surprisingly, we got results that are slightly worse than those reached using one quarter of the trigraphs. After all, the average number of trigraphs involved in these last two experiments—68 and 64—are similar, so we expected an improvement of the authentication accuracy because of the use of trigraphs typed more than once. However, the use of such trigraphs apparently does not help to improve the accuracy of the distance measure, and we may learn from this an important lesson. If repeated trigraphs are not useful, they can be avoided as much as possible, in this way limiting the length of the sample text needed to reach a sufficient level of authentication accuracy. The sample text used in our experiments was chosen only because it is very famous in Italy. It contains about 350 different trigraphs, but only 272 on the average are shared between two samples under comparison, because of typing errors. A carefully chosen text can be made of about 300 characters, and can contain a similar number of different trigraphs, in this way providing an authentication accuracy similar to the one reached in our best outcomes, but with a text much shorter. Of course, in this case, samples cannot contain typing errors. However, we do not need to reject samples because of typos: an on-line procedure can be used to notice the error as soon as it is made, and ask the user to correct it.

[14] This is roughly one quarter of the number of trigraphs shared on the average by two complete samples. In fact, about one quarter of the trigraphs that make up the original sampling text occur more than once.



(a) (b)

(c)

Fig. 4. (a) A graphical representation of the mean distances between samples of the same and different users. (b) Distributions of the distances of all the samples. (c) Distributions of the distances of samples from the same user.

7.2 Experimental Properties of Distance Measure used in the Experiments

One observation about some experimental properties of the measure of distance used in the experiments can also be useful. Consider the 330 samples collected in our experiments. The distances between any two samples of the same user have a mean of 0.363964 and a standard deviation of 0.023224, whereas, for distances between any two samples from different users, we have a mean of 0.55993 and a 0.024556 standard deviation. The situation is graphically depicted in Figure 4(a).

These values account well for the good performances of our approach, and provide support for what was observed in Section 2. We may expect two typing samples from the same user to be more similar than two samples from different users. In other words, when we compare a sample S with another sample S′ both from the same user U, we may expect S to show a similar ordering, with respect to the ordering of S′, and hence they will have a small distance measured in terms of their relative disorder. On the other hand, if S belongs to another user, we may expect it to be almost randomly sorted[15] with respect to a sample of U.

[15] Clearly, a complete relative randomness cannot be assumed: for example, a particular trigraph can be very hard to type, and any typist would probably type it slowly. Nonetheless, relative randomness of samples increases when comparing samples from different users, as also shown in Figures 4(b) and 4(c).

Table V. Mean Distances for Digraphs, Trigraphs, 4-graphs and 6-graphs

	digraphs	trigraphs	4-graphs	6-graphs
mean distance between any two samples from the same user	0.31651	0.363964	0.379610	0.399062
mean distance between any two samples from different users	0.509375	0.559930	0.569073	0.574657
meandiff	0.192865	0.195966	0.189463	0.175595
average number of shared n-graphs between samples	160	272	254	172

But as we saw in Section 2, a randomly generated array will have, with high chances, a degree of disorder falling in the interval [0.5–0.75], precisely like the mean of all the distances of two samples of different users, as measured in the experiments. Such situation in depicted in Figure 4(b). The bar graph shows the distribution in the interval [0–1] of all the distances between any two samples gathered in the experiments. Figure 4(c) shows the distribution of all the distances between any two samples from the same user. The shape of the distribution in Figure 4(b) conforms perfectly with the expected distribution of the disorder of arrays of about 272 elements, as noted in Section 2. Moreover, most of the distances between two samples from different users gather in the interval [0.5–0.75], whereas most of the distances between two samples from the same user concentrate in the interval [0.3–0.5]. Finally, as we already noted, the ratio between the number of arrays whose disorder is greater than 0.5 and the number of arrays whose disorder is smaller than 0.5 increases together with the size of the arrays (see Figures 2(a) and 2(b). This gives an additional explanation of the fact that the longer the text used to generate the samples, the more accurate the user authentication.

7.3 On the Use of Digraphs, Trigraphs, and *n*-Graphs

One may wonder how much of the authentication accuracy of our approach is due to the use of trigraphs instead of digraphs, and how the use of 4-graphs or more would affect the outcomes. We have tested our method using digraphs, 4-graphs and 6-graphs, and trigraphs was the choice that proved to perform better. An experimental evidence of this can be provided by computing the mean distance between any two samples of the same user and the mean distance between any two samples from different users, as we did in the previous section, even for digraphs, 4-graphs and 6-graphs. In particular, it is important to consider the distance between the two means, and the average number of n-graphs shared by any two samples under comparison. These values are shown in Table V (we do not report the standard deviations, which are quite similar for all of the computed means). Also, in order to simplify the discussion, let us define *meandiff* = mean distance between any two samples from different users − mean distance between any two samples from the same user.

Intuitively, the larger *meandiff*, the higher the ability of the distance measure to discriminate among users, and, from Table V, we see that the largest value for *meandiff* is reached using trigraphs. Actually, very similar values are reached with digraphs, but with an important difference: the average number of digraphs shared by two samples under comparison is smaller than the corresponding number of trigraphs. This happens because in any language there are more legal trigraphs than digraphs, and any text sufficiently long has more repeated digraphs than trigraphs.[16] But, as we have seen in Section 7.1, the larger the number of n-graphs used the higher the authentication outcomes, whereas the presence of n-graphs occurring more than once is apparently not useful.

When moving to longer n-graphs, two things contribute to worsen the accuracy of the distance measure. (1) The presence of typing errors decreases the number of shared n-graphs, and the higher the value of n, the smaller the number of shared n-graphs.[17] (2) In general, the duration of longer n-graphs is less stable since they are made of more keystrokes. As a consequence, the comparison of the duration of the same n-graph occurring in two different samples provides outcomes less accurate. As a limit situation, consider the case of using n-graphs where n is the length of the sample text. Then, there is only one n-graph available (if we assume no typing errors occurred), and discrimination between users is left to compare the time required to type a sample. Clearly, in this case, users with similar typing speed could be easily confused, even if they have quite different typing rhythms. Moreover, the freedom of users to stop when they want, and to correct errors, would add further confusion.

7.4 On the Set of Volunteers Involved in the Experiments

As observed, 154 volunteers were involved in our experiments. Among them, different habits and typing skill could be found. All four secretaries working in our department provided five samples, hence impersonating legal users. All these women attended a secretarial school. Moreover, a few other people involved in the experiments had professional typing skills. Quite interestingly, in all of the authentication experiments involving the whole sample phrase (outcomes of Table III), none of the samples belonging to such skilled typists was erroneously recognized to belong to another user.

At the opposite end of the typing skills, there are a few individuals who have very low typing skills. Even if they use computers as part of their job, they are slow to type, and normally use only the forefinger of each hand. In particular, two of them use only the right forefinger. Such basic typing skills appear to be the most easy to reproduce by potential impostors. In order to check this, we asked a few people to type the sample text using only the right forefinger, as these two individuals do. None of the samples produced in this way were erroneously attributed to one of the two unskilled individuals.

[16] For example, the title of this article is made of 39 different digraphs, with five of them occurring twice (if we consider *Ke* and *ke* to be equivalent), whereas it has 43 trigraphs with no repetitions.

[17] As an example, the two words *America* and *Ameriga* share three trigraphs, two 4-graphs, and no 6-graphs.

Nonetheless, individuals with slow typing speed seem to be those with higher chances to be attacked, and, in fact, most of the authentication errors occurred in our experiments involved samples of people typing slowly. There is also an experimental evidence of this phenomenon. For every pair of the 44 legal users, we have computed the mean of the distances between any two samples of the two users (let us call such value *mdu*).[18] There are 946 such pairs of users, with the *mdu* of each pair varying from 0.315656 to 0.752554. The smallest of such values come from pairs of slow typing users, whereas the largest values come from the comparison among samples of very fast and very slow users. Moreover, even the *mdu* between two fast typists is high, normally larger than 0.6, and this is true also for the professional typists involved in our experiments. Hence, from the above values, it appears that it is easier for a slow typist to impersonate another slow typist than for a fast typist to impersonate another fast typist.

7.5 Scalability

Another important issue about our method concerns its scalability. In general, the acceptance or rejection of an access attempt depends on who else is a legitimate user. In the worst case, if a new user has a profile that is too close to another user, both could be rejected. Clearly, the larger the number of legal users of a system, the higher the chances that two legal users have similar profiles, and one may wonder how the space of profiles fills in as the number of users increases.[19]

Only extensive experiments could properly answer the above question, but some general remark can be made on the basis of the samples of the 44 legal users of our experiments. Let us consider m(A) (as defined in Section 6.3) as a sort of number representative of the way A types the sample text: m(A) gives as an idea of the distance we may expect between two typing samples provided by user A. Consider also the *mdu* as defined in the previous section. For two users A and B, mdu(A,B) gives us an idea of the distance we may expect between any sample of A and the model of B, or vice-versa.[20]

In general, the smaller mdu(A,B), the higher the chances that users A and B can be both rejected. However, the true meaningful information lies in the relation between m(A) and mdu(A,B). When $\mathrm{m(A)} \simeq \mathrm{mdu(A,B)}$ then samples from A and B cannot be distinguished any longer. Such a situation never occurs for any of the 946 pair of users involved in our experiments and, in the worst case, mdu(A,B) is about 20% larger than m(A) and m(B).

As another evidence of the scalability of our approach, we have also repeated the classification experiment of Section 5, but adding to the 220 samples of 44

[18] For example, for two users A and B, each one providing two samples, we have $\mathrm{mdu(A,B)} = (\mathrm{d(A1,B1)} + \mathrm{d(A1,B2)} + \mathrm{d(A2,B1)} + \mathrm{d(A2,B2)})/4$.

[19] However, one may also observe that the larger the number of legal users, the lower the chances that an impostor's sample is by accident closer to the attacked user's model than to any other system's user, as discussed in Section 6.2.

[20] In fact, for example, for two users A and B with two samples each one we have:

$$\begin{aligned}(\mathrm{md(A,B1)} + \mathrm{md(A,B2)})/2 &= ((\mathrm{d(A1,B1)} + \mathrm{d(A2,B1)})/2 + (\mathrm{d(A1,B2)} + \mathrm{d(A2,B2)})/2)/2\\ &= (\mathrm{d(A1,B1)} + \mathrm{d(A1,B2)} + \mathrm{d(A2,B1)} + \mathrm{d(A2,B2)})/4 = \mathrm{mdu(A,B)}.\end{aligned}$$

users also the samples of the 110 persons who provided one sample.[21] Hence, now each of the 220 samples must be classified among 154 users, instead of 44. Nonetheless, all 220 samples are still correctly classified.

Finally, consider again Figures 4(b) and 4(c). Their partial overlapping may suggests a potential overlapping of two users' models of our experiments. Overlapping of Figures 4(b) and 4(c) happens when d(X1,X2) $\simeq$ d(Y1,Z1), for some users X, Y and Z. However, overlapping of two users' models may happen only if X = Y (or X = Z), and only if this happens for most of the samples of X and Y (or Z).

In our approach, scalability is also related to the number of comparisons needed to authenticate an incoming sample. In fact, the method described in Section 6.3 requires a new sample X to be compared against all users' profiles samples. When the number of legal users in the system is very large, this operation becomes computationally unfeasible. A solution to this problem is just to partition the whole set of users into subsets of manageable size. When an incoming sample X is claimed to belong to a legal user A, X is compared only against the profiles of users in the subset containing A.

Partitioning can be done randomly, or it can be based on the value of the *mdu* of any two users of the system. At one extreme, one may choose every partition to contain only users that have a large *mdu* between each other. System's FAR would benefit from such solution, since there would be less chances that a user's sample is erroneously recognized as belonging to another user in the same partition (recall the observation made at the beginning of this section about users with similar profiles). At the other extreme, grouping together users that have a small *mdu* between each other will improve system's IPR. In fact, an impostor pretending to be user A will have his/her sample compared against a set of users having typing habits similar to A. Hence, that sample will have less chances to be classified as belonging to A. A trade-off between the two choices will probably provide the best system performance. Of course, even the number of users in each partition affects the authentication accuracy of the system. For example, a number of users and samples in each partition similar to the one adopted in our experiments will very likely provide a similar performance. Much larger partitions may be adopted in order to lower the system IPR, as long as the computational costs remain acceptable.

A smart partitioning like the one described above is computationally expensive, since *mdu* must be computed for any pair of users in the system. However, such computation must be done only once, and can be performed offline. When a new user U is introduced in the system, his/her *mdu* with respect to all existing users must be computed in order to put U into the right partition.

8. OTHER APPROACHES TO KEYSTROKE ANALYSIS

It is difficult to clearly compare the various biometric systems based on keystroke analysis because authors have adopted different approaches to the

[21] Clearly, for a user A having a model that contains only one sample A1, the mean distance of a sample to be classified X from A's model is computed as md(A,X) = d(A1,X).

experimental setting. There is however a well-established set of works that can be considered as the reference literature.

Apart from the pioneering work found in Gaines et al. [1980] (which is difficult to evaluate, since it involved only seven individuals), one of the first, and most cited research is found in Umphress and Williams [1985], later improved in Leggett and Williams [1988]. The approach compares samples using digraph latency, and the experimental setting includes 36 individuals who provide twice the same text of 537 characters, with a delay of at least one month. The first sample is used as a model of the user, and the second sample is the one the individual provides to be authenticated. The second sample of each individual is also used to "attack" every other individual. As a consequence, there are 36 legal connection attempts and 1260 attack attempts (each individual pretends to be one of the other 35 ones). The outcomes are an FAR of 5.5% and an IPR of 5.0%. The experimental setting of this research is pretty similar to the one adopted in this paper, because the same text is adopted for all of the users, and because typing errors are allowed.

In Joyce and Gupta [1990], again digraph latencies are used and 33 legal users are asked to type, in the same typing session, eight plus five times the same sample that, for each user, is made of his/her name, surname, login name and password. The first eight samples are used as a model of each user, the remaining five are used to test the system. This turns out into 165 legal connection attempts, with an FAR of 16.36%. To test the IPR of the approach, six legal users are chosen randomly, and other 27 individuals are asked to type five times the sample string of the legal users. This allows to simulate a total of 810 attacks brought five times to the six legal users by 27 individuals, with a resulting IPR of 0.25%. The experimental FAR of this approach is quite high, in spite of the fact that all the samples of each legal user are collected at the same time, and that, moreover, each user should be well familiar with his/her own sample string. On the other hand, this approach adopts thresholds to accept/reject samples, and improving the FAR would in general imply a worsening of the corresponding IPR.

Thresholds learned on the basis of available samples of legal users and impostors are also used in Bleha et al. [1990]. Digraph latency is used by a Bayes classifier and typing models of the users are produced using their name, surname, and a common fixed text of 31 characters. The training of the authentication system goes on for a few weeks using the samples provided by valid and invalid users, in order to adjust the classification thresholds. Samples containing typing errors are discarded. Finally, the tailored system is tested by ten legal users who already participated to the training phase, and by 22 impostors (it is unknown if these individuals also participated to the training phase). The authentication system provides an FAR of 3.1% (for 380 valid connection attempts) and an IPR of 0.5% (for 220 impostors attacks.).

Another much cited research in keystroke analysis is the one found in Brown and Rogers [1993]. In this approach, keystroke duration and latency are used, and authentication is made in three different ways, using Euclidean distance and two different kinds of neural networks trained with samples of the users. Typing samples containing errors are rejected and, moreover, also correct samples from a given user not similar enough to the other from the same user are discarded when forming a model of that user. The same set of experiments is repeated over two different groups of users. The first group is made of 21 users, the second group includes 25 users. For all of the users, a sample is made of his/her own name and surname, on the average between 15 and 16 characters. To form a training set of samples (to be given in input to the neural networks), each user of the first and second group provided on the average about 29 and 47 typing samples of his/her name, to be used as positive examples to the neural networks. The average elapsed time between the gathering of one sample and the next one, for each user, is unknown. Each user also typed once or twice the name of the other users, in order to provide negative examples to the neural networks. To test the three authentication methods, each legal user is attacked twice by 15 impostors, for a total of 610 and 750 attack attempts for the two groups. Each legal user provides from 10 to 15 further samples to be used as legal connection attempts, for a total of 241 and 330 legal attempts for the two groups. The authors claim an impressive IPR = 0%. However, this is obtained simply by setting in advance the thresholds of the three authentication methods (with respect to the available samples) in order to reject all the impostors samples, and then seeing what happens to the legal connection attempts. Using the Euclidean distance analysis technique, they get an FAR of 14.9% and 23.6% for the first and second group of users, respectively. Using a backpropagation neural network, they get an FAR of 17.4% and 40%. Using a partially connected backpropagation network, they get an FAR of 12.0% and 21.2%. By picking the best FAR outcomes (i.e., the method that performs best for the available samples of every user) for each user of each group, they can claim an FAR of 4.2% for the first group of users and an FAR of 11.5% for the second group. The outcomes of these experiments are very good, especially because of the very short text used for each sample. However, the results are obtained at particular conditions, and appear to be tailored to the specific set of available samples. Hence, it is not clear if such performance could be maintained in real applications.

A similar line of research is found in Obaidat and Macchairolo [1994] and Obaidat and Sadoun [1997a, 1997b]. In particular, in Obaidat and Sadoun [1997b], keystroke duration and latency are used, together with different kinds of neural networks. Fifteen users provided their login name 225 times each day (on the average, seven characters per sample) over a period of 8 weeks. This large set of samples is split to provide a training and testing set. Fifteen individuals attack each of the 15 users 15 times. All the attacking samples of each attacker are produced in the same session. It is unclear whether the 15 attackers are the same 15 users who provide the legal samples. A perfect FAR = IPR = 0% is achieved, but, clearly, at very special conditions: each legal user has probably provided his own login name a few thousand times in order to build his profile, whereas an attacker only types the same text 15 times. It would be interesting to test the IPR of this approach in a situation where attackers can practice to type someone else's login name as many times as the legal owner of that login name.

As a common feature, many of the systems described in this section strive to work with very short sample texts. Hence, one may well note that it is unfair to compare the outcomes of such systems with those of our approach, which uses a much longer text. However, one may also observe that the outcomes of our method do not rely on any form of continuous and heavy training of the legal users to type the pass phrase (training that is however not allowed to the attackers in the above systems), and that no form of tailoring of the authentication method to each user's profile is adopted.

9. APPLICATIONS

Proposing the authentication method described in this paper as a plain substitute of password-based systems is clearly unrealistic. After all, passwords work well for most applications, and require a very limited number of characters to be typed. Nonetheless, the use of keystroke analysis can be well motivated in many situations, and some of them are discussed in this section.

9.1 Forgotten Passwords and Strong Authentication

Access Control Systems based on passwords work very well for most security applications, but they suffer an obvious and well-known problem: passwords can be forgotten. This problem is especially endemic over the Web, where a myriad of sites provide all kinds of reserved services, whose access is normally controlled through passwords. The most common of such services, used probably by millions of people everywhere, is a free e-mail account, sometime offered together with some disk space for personal Web pages. Especially if the connection to such sites is infrequent, users tend to forget the passwords they chose, so that some form of automatic protocol is normally available on the site to help users to retrieve the lost password and regain access to the service. In many cases, password retrieval works by answering *personal questions* set by the user at the very first registration at the site, and whose answer is supposedly only known to the user. It is well known that the proper answers are often forgotten before the password they should help to remember, or are so easy that also others may know them. Of course, users may choose to sign up anew on the site, but this is hardly an acceptable solution in case of lost e-mail account passwords. In such situations, an alternative authentication method based on keystroke analysis can be provided by the site to allow a registered user to regain access to his/her account on the site in a few minutes, with a minimum burden. That is, users not remembering their passwords can choose to be authenticated through the method described in this article, via a Web-based client-side application gathering the keystroke timing to be sent to the server for the authentication step.

Though rarely, even a job's account passwords can be forgotten. Within local networks, in the case of Internet providers, or when connecting from home to the office server late at night, losing one's own password means contacting a system administrator (who is not always immediately available), and undergoing a new authentication procedure, which may take time. In all cases, the user's account may remain unaccessible for a period of time varying from a few hours to days. Even in such cases, users may be willing to undergo an alternative authentication procedure based on our approach to keystroke analysis. To guarantee a high level of security, sample phrases even longer than the one chosen for our experiments can be used, typing errors will have to be avoided or corrected, and more accurate users' profiles will be needed. Nonetheless, in many cases, this would by far more acceptable than having to wait for a system administrator to take care of the problem.[22] As proved by our experiments, our method would work well even in all cases of remote connections through modems.

For all the above applications, we believe our approach to be safer and more accurate than the techniques described in the previous section. First, our method is not based on any form of training to type the adopted sample text, and this is important for an authentication process that would be used only in the rare cases of a forgotten password. Second, it requires a relatively long sample phrase. As observed, this is a relatively harmless drawback, since it would be typed rarely, and it has two major advantages: (1) it takes some time to be entered, so that potential intruders cannot attempt an attack many times, (2) attackers have little chances to get used to type the pass phrase, while it could be easily possible with sample texts of a few characters. With our method, the desired level of security can be achieved by choosing a sample phrase of appropriate length. Moreover, sample texts can be chosen to be different for each user, so that it would also be difficult for an attacker to guess them, as it would be easily the case for sample texts consisting in users' affiliation and/or names.

Clearly, our authentication system could be used also *in conjunction* with passwords, in all those specific applications where a very high level of security is needed. An even higher level of security can be reached by choosing the sample text itself to be the secret pass phrase to know in order to be authenticated. Though much longer than common passwords, a secret pass phrase can still be kept in mind if it consists of plain text, such as the beginning or ending parts of novels, or poems, and so on. Secret pass phrases would have to be entered only in safe places protected from any kind of spying, since it would be too awkward to type long texts without having the entered characters displayed, as in the case of common passwords.

9.2 Identity Confirmation

Not only passwords can be forgotten: they can be stolen or cracked. This may happen in many ways, including user's negligence, but the consequence is more or less always the same: an impostor will use someone else's account. Apart from extreme cases, such as an intruder entering an account, stealing or removing everything and leaving, impostors try to go unnoticed as long as possible, while using the intruded account and system resources for their fraudulent aims. Of course, such impostors should be spotted as soon as possible, with an acceptable level of accuracy, and immediately disconnected.

Intrusion Detection systems [Axelsson 2000a] have two main ways to realize that an intrusion is under way or has occurred recently: (1) some users or some known user processes behave in a way that is clearly unusual, for example, a

[22] Moreover, if concerns are raised about the security of accounts entered through keystroke analysis authentication, access may be granted in a restricted mode, for example, read-only.

secretary starts running *awk* and *gcc*; (2) a typical attack pattern is recognized, for example, some user reads a password file or attempts to delete system logs. In the first case, we speak of *anomaly detection*, while the second objective is defined as *misuse detection*. Unfortunately, both approaches are prone to errors. In particular, misuse detection systems are useless in presence of novel patterns of attacks. On the other hand, anomaly detections systems can be effective against unknown forms of attacks (since do not need any form of a priori knowledge about specific forms of intrusions) but they tend to generate more false alarms as a consequence of possible changes in legal users' habits. False alarms are an endemic problem of intrusion detection systems [Axelsson 2000b; McHugh 2000], and the false alarm rate should be kept small to avoid bothering both legal users and system administrators.

There are, of course, clear anomalies and misuses that can be easily noticed, such as someone copying a password file at 2 AM from a secretary account. However, in many other cases, some *strange* user behavior may have been detected, but a clear evidence on an intrusion has not yet been reached. In all such situations, in order to avoid false alarms and at the same time avoid intrusions, *suspicious* cases can be handled by asking the user under observation to provide a further evidence confirming his/her identity. Our authentication method can be used to this aim, if a typing model of each legal user of the system is available. The individual using an account that is raising a possible alarm may be asked to enter a new typing sample to be checked against the typing model of the legal owner of the account. An identity verification failure would result in an immediate disconnection of the individual and blocking of the account, to be reactivated only after reregistration and password change of the owner of the account. For the same reasons mentioned in the previous section, our method would be well suited for this application that hopefully, will be needed rarely.

A last remark concerns scalability of computational costs. For the applications described in this section and in the previous one, a prompt response to an incoming new sample is not mandatory, as it would be in the case of a user waiting to be authenticated to enter his/her account. When a password has been lost or when an intrusion is suspected, devoting say one minute to the authentication task appears quite acceptable, so that even dealing with a large number of users' profiles is possible. Nonetheless, if needed, computational costs can be limited through the partitioning technique described in Section 7.5.

9.3 User Identification and Tracking over the Internet

Being able to offer personalized services, catalogs, selected products and targeted advertising is one of the most important aspects that Websites (especially the commercial sites) can exploit to improve their success (and often their incomes) on the Net [*Commun. ACM* 2000]. This means being able to record and analyze users' habits, choices and preferences when visiting a site, so that some form of adaptation to visitors' needs can take place [Perkowitz and Etzioni 2000a, 2000b]. However, site personalization loses much of its potential if users cannot be identified on their returning to the site. User identification and tracking over the Internet is commonly achieved through cookies or IP numbers, but these methods have drawbacks. In fact, user identification through his/her host IP number becomes very difficult if the user can connect from different hosts, if the connection goes through a proxy, in case of dynamic IP numbers, or in case of multiple users navigating the Net from the same server. Similarly, cookies become useless in the case of connections from different hosts. Moreover cookies can be disabled by the browser or firewall configuration, and can even get lost [Pitkow 1997]. On the other hand, a given computer may be shared by multiple individuals, making the use of its IP number and cookies essentially meaningless.[23] Finally, Web pages caching, used to improve efficiency, prevents the originating server of cached pages from identifying who is actually downloading such pages [Davison 2001; Pitkow 1997; Burton 2001].

An alternative form of user identification can be based on our approach to keystroke analysis, again with the keystroke timings gathered by a Web-based client-side application and then elaborated on the server where the user has to be identified. Clearly, such form of identification would work independently of the host used by the user to connect. Moreover, for the task of user identification, the requirement for more than a few bytes of sample text of our approach appears appropriate. Identifying a user through the way he/she types, means in fact being able to classify his/her sample text among a set of available profiles, and results of Section 5 show that our method works well to this aim, and that it can be scaled (Section 7.5). Admittedly, it is unclear *to what extent* it can be scaled, and it is well known that user identification becomes a very difficult task when the identification must be done among thousands of individuals [Ashbourn 2000b]. Clearly, even the computational costs needed to identify a user among very many individuals increase significantly, since all users' profiles must be taken into consideration in order to classify a new sample. The problem can be partially mitigated by the fact that some time may be available to perform the identification task, if we assume users spend time visiting a given site. Moreover, our approach can also be combined with some of the tracking techniques mentioned above in order to limit computational costs. For example, when multiple individuals navigate from the same server or share the same PC, IP numbers and cookies are still useful to select the subset of users that are known to connect from that host. Our classification method would then be used on that subset to identify the actual connecting user.

Quite interestingly, our method shows a certain level of identification accuracy even in the case of different sample texts typed by the users to be identified. To check this, we have done the following experiment with the 220 samples provided by the 44 users who typed the sample text five times. Each sample X has been split into two new samples, Xa and Xb, containing respectively the digraphs of the first and second half of the original sample phrase (Section 4.1). For each pair of samples Xa, Yb, the distance d(Xa,Yb) has been computed. Note that, in general, the computation of the distance of two typing samples made of different texts is possible, as long as they share some digraphs. In fact,

[23] Think, for example, to a family connecting from home to one of the many Web bookstores now available. Of course, a different offering of books, movies and videogames would be appropriate for each member of the family.

every language allows only for a finite numbers of legal digraphs (i.e., digraphs coming from words of the adopted language) and hence a comparison of typing samples of different texts can be made if the samples share a number of digraphs sufficiently large.[24] In our case, each "a" and "b" sample comes from a text of about 683/2 characters and, on the average, two "a" and "b" samples share 82 digraphs. A classification of each "b" sample has been attempted using its computed distance with the "a" samples, resulting into a correct identification of the user who typed that "b" sample of about 90%. This feature can be particularly useful to identify returning users of the many sites that provide, beside other services, dedicated chat lines or forums. Often, such services can be accessed without any form of registration, so that it is difficult to track users. However, visitors sending messages through chat lines and forums can produce a large amount of typed text, that can be recorded and used to identify them on their return to the site, even when connecting from different hosts and/or with different nicknames.

We conclude by observing that, whatever technique is adopted for user identification and tracking over the Web, it may raise some concerns about user's privacy [Volokh 2000]. At the very least, users should be made aware that they are under observation, and should be allowed to disable the tracking process, if desired [Vora et al. 2001]. In the case of keystroke analysis, typed text other than common pass phrases will have to be blurred, and only stored in terms of *n*-graphs and the corresponding timing information needed for the identification process.

10. CONCLUSION

In this article, we have described a new biometric measure of the typing characteristics of individuals. The measure has been tested on a set of volunteers larger than in many other experiments described in the literature, performing very well to authenticate legal users and to reject impostors: on the average, we got a 4% False Alarm Rate and an Impostor Pass Rate of less than 0.01% (i.e., less than one successful attack out of 10,000 attempts). Unlike other methods found in the literature, our approach allows typing errors, uses the same sampling text for all the individuals, and requires a small number of samples to form the typing model of a user. Finally, the method works well without any form of overfitting, and even through remote connections. An authentication system using the approach described in this article does not need any specific tuning, nor a learning phase to work with a particular set of legal users of the

[24] In this experiment, digraphs have been used because in any language there are less legal digraphs than trigraphs. Hence, when computing the distance of two typing samples of different texts, such samples will, in general, share a larger number of digraphs than trigraphs, so increasing the meaningfulness of the computed distance (see also Section 7.3). The above can be understood by observing that: (1) at least one legal trigraph can be built from a legal digraph, but often more trigraphs are allowed (such as in the case of *th* that can be followed by any vowel, in this way turning into more legal trigraphs in English); (2) as a consequence of (1), on the average a given digraph has higher chances to occur in two different sentences with respect to one of the trigraphs stemming from the same digraph.

system. This tuning is however possible, in this way increasing the ability of the system to authenticate legal users and reject impostors.

The text used in our experiments is too long to be used to replace a password based authentication system, but its length is acceptable for other practical applications, as suggested in the previous section. Moreover, there is evidence that the adopted sample phrase can be shortened sensibly through a careful text choice, requiring errors to be corrected, and using a more accurate and precise timing sampling. Also, the duration of trigraphs adopted in our experiments is a pretty rough measure, and our method could benefit from the use of more precise information such as, for example, digraph latency, as discussed in Mahar et al. [1995] and Obaidat and Sadoun [1997].

Keystroke dynamics is the most obvious kind of biometrics available on computers, but it has not yet led to real security applications, if compared to other biometric measures. However, we believe keystroke analysis can be a practical tool to help implementing access control systems to computer resources and other related applications, and our study represents a contribution to this aim.

ACKNOWLEDGMENTS

Thanks to the volunteers of our department who contributed to our research. Thanks also to the anonymous reviewers who suggested many improvements to the article.

REFERENCES

ASHBOURN. J. 2000a. *Biometrics: Advanced Identity Verification. The Complete Guide*. Springer-Verlag, London, Great Britain.

ASHBOURN, J. 2000b. The distinction between authentication and identification. Paper available at the Avanti Biometric Reference Site. (homepage.ntlworld.com/avanti)

AXELSSON, S. 2000a. Intrusion detection systems: A taxonomy and survey. Tech. Rep: 99-15. Dept. Computer Engineering, Chalmer University of Technology, Sweden, March. Paper available at www.ce.chalmers.se/staff/sax/taxonomy.ps.

AXELSSON, S. 2000b. The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Sec. 3*, 3, 186–205.

BLEHA, S., SLIVINSKY, C., AND HUSSEIN. B., 1990. Computer-access security systems using keystroke dynamics. *IEEE Trans. Patt. Anal. Mach. Int. PAMI-12*, 12, 1217–1222.

BROWN, M. AND ROGERS, S. J. 1993. User identification via keystroke characteristics of typed names using neural networks. *Int. J. Man-Mach. Stud. 39*, 999–1014.

BROWN, M. E. AND ROGERS, S. J. 1996. Method and apparatus for verification of a computer user's identification, based on keystroke characteristics. Patent Number 5,557,686, U.S. Patent and Trademark Office, Washington, D.C., Sept.

BURTON, M. C. 2001. The value of web log data in use-based design and testing. *J. Comput. Med. Commun. 6*, 3. Also available at: www.ascusc.org/jcmc/vol6/issue3/burton.html

Commun. ACM, Special issue on Personalization. Volume 43, Number 8. 2000.

DAVISON, B. 2001. A web caching primer. *IEEE Internet Comput. 5*, 4, 38–45.

FURNELL, S., MORRISSEY, J., SANDERS, P., AND STOCKEL, C. 1996. Applications of keystroke analysis for improved login security and continuous user authentication. In *Proceedings of the Information and System Security Conference.* pp. 283–294.

GAINES, R., LISOWSKI, W., PRESS, S., AND SHAPIRO, N. 1980. Authentication by keystroke timing: Some preliminary results. Rand. Report R-256-NSF. Rand Corporation.

GARCIA, J. 1986. Personal identification apparatus. Patent Number 4,621,334, U.S. Patent and Trademark Office, Washington, D.C., Nov.

JOYCE, R. AND GUPTA, G. 1990. User authorization based on keystroke latencies. *Commun. ACM 33*, 2, 168–176.

LEGGETT, J. AND WILLIAMS. G. 1988. Verifying identity via keystroke characteristics. *Int. J. Man-Mach. Stud. 28*, 1, 67–76.

LEGGETT, J. WILLIAMS, G., AND USNICK, M. 1991. Dynamic identity verification via keystroke characteristics. *Int. J. Man-Mach. Stud. 35*, 859–870.

MAHAR, D., NAPIER, R., WAGNER, M., LAVERTY, W., HENDERSON, R., AND HIRON, M. 1995. Optimizing digraph-latency based biometric typist verification systems: inter and intra typist differences in digraph latency distributions. *Int. J. Human-Comput. Stud. 43*, 579–592.

MCHUGH, J. 2000. Testing intrusion detection systems. *ACM Trans. Inf. Syst. Sec. 3*, 4, 262–294.

MONROSE, F. AND RUBIN, A. 1997. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*. ACM, New York, pp. 48–56.

REITER, M. K., MONROSE, F., AND WETZEL, S. 1999. Password hardening based on keystroke dynamics. In *Proceedings of the 6th ACM Conf. on Computer and Communications Security* (Singapore), ACM, New York, pp. 73–82.

OBAIDAT, M. S. AND MACCHAIROLO, D. T. 1994. A multilayer neural network system for computer access security. *IEEE Trans. Syst. Man, and Cybernet. Part B: Cybernet. 24*, 5, 806–812.

OBAIDAT, M. S. AND SADOUN, B. 1997a. A simulation evaluation study of neural network techniques to computer user identification. *Inf. Sci. 102*, 239–258.

OBAIDAT, M. S. AND SADOUN, B. 1997b. Verification of computer users using keystroke dynamics. *IEEE Trans. Syst. Man, and Cybernet. Part B: Cybernet. 27*, 2, 261–269.

PERKOWITZ, M. AND ETZIONI, O. 2000a. Adaptive web sites: Conceptual framework and case study. *Artif. Int. 118*, 1, 2, 245–275.

PERKOWITZ. M. AND ETZIONI, O. 2000b. Adaptive web sites. *Commun. ACM 43*, 8, 152–158.

PITKOW, J. 1997. In search of reliable usage data on the WWW. In *Proceedings of the 6th International WWW Conference* (Santa Clara, Calif.). Also available at: www.parc.xerox.com/istl/groups/uir/pubs.

POLEMI, D. 2000. *Biometric techniques: review and evaluation of biometric techniques for identification and authentication, including an appraisal of the areas where they are most applicable*. Report prepared for the European Commission DG XIII-C.4 on the Information Society Technologies (IST) (Key action 2: New Methods of Work and Electronic Commerce). Report available at: www.cordis.lu/infosec/src/stud5fr.html.

VOLOKH, E. 2000. Personalization and Privacy. *Commun. ACM 43*, 8, 84–88.

VORA, P., REYNOLDS, D., DICKINSON, I., ERICKSON, J., AND BANKS, D. 2001. Privacy and Digital Rights Management. *World Wide Web Consortium Workshop on Digital Rights Management for the Web*. Also available at: www.w3.org/2000/12/drm-ws/pp/hp-poorvi.html.

UMPHRESS, D. AND WILLIAMS, G. 1985. Identity verification through keyboard characteristics. *Internat. J. Man-Mach. Stud. 23*, 263–273.

YOUNG, J. R. AND HAMMON, R. W. 1989. Method and Apparatus for Verifying an Individual's Identity. Patent Number 4,805,222, U.S. Patent and Trademark Office, Washington, D.C., Feb.

Received August 2001; revised February 2002 and May 2002; accepted May 2002