Dependable Computing

John Rushby Editor

Identity Authentication Based on Keystroke Latencies

The variables that help make a handwritten signature a unique human identifier also provide a unique digital signature in the form of a stream of latency periods between keystrokes. This article describes a method of verifying the identity of a user based on such a digital signature, and reports results from trial usage of the system.

Rick Joyce and Gopal Gupta

Computer systems are now used in almost all aspects of business and commerce and many businesses rely heavily on effective operations of their computer systems for their business to succeed. For computer systems to be effective, they must be secure so that information stored in them is accessible only to authorized users.

Computer security usually involves several components:

- physical security of the computer installations so that unauthorized persons may not enter the installations.
- identification, authentication and authorization mechanisms to ensure that persons accessing the computers remotely are allowed access to the systems only if they are authorized to have such access. Use of login names and passwords is the most common mechanism to control user access to computer systems although some sensitive installations require that the user insert a user identification card in specially designed user terminals.
- physical security of computer terminals is also used in some sensitive computer systems. This usually restricts the user to access the computer system only through one of the designated terminals that are placed in physically secure locations.

This article deals only with user authentication. Methods for verifying the identity of an individual can be divided into four classes:

(1) *objects* in the possession of the individual, such as keys, id cards, passports, etc.
(2) *knowledge* that the person has, such as lock combination, password, PIN number, etc.
(3) *actions* such as signature or patterns of behavior.
(4) *physiology* such as the physical description, fingerprints, retinal pattern, voice pattern, etc.

Computer systems commonly use the first two of these categories, e.g., possession of keys to the building along with a valid username/password, or possession of a bank card along with knowledge of its corresponding PIN number. Some work has been done in the last category but as of yet these techniques require expensive, specialized hardware and software. To date, the "actions" category has been largely ignored.

Use of login names and passwords is the most commonly used mechanism for static identification and authentication. The apparent ease with which hackers have been able to access many systems that were considered secure clearly indicates the inadequacy of the password mechanism for verifying identity and therefore there is a need for other, more reliable, security measures.

Society has relied on the written signature to verify the identity of an individual for hundreds of years. The complexity of the human hand and its environment make written signatures highly characteristic and difficult to forge precisely. In current computer systems, the signature has been replaced by a username/password pair (coupled with encryption schemes) for static identification and authentication. One problem with this scheme is that it relies entirely on the "knowledge" category of authentication techniques, and has abandoned the information contained in the "actions" category.

The handwritten signature has a parallel on the keyboard. The same neurophysiological factors that make a written signature unique are also exhibited in a user's typing pattern. When a person types on a keyboard, he/she leaves a digital signature in the form of keystroke latencies (the elapsed time between keystrokes). For well-known, regularly typed strings this signature can be quite consistent.

© 1990 ACM 0001-0782/90/0200-0168 $1.50

The idea of using keyboard characteristics in identifying and verifying individuals is not new and some products that use such characteristics have been known to be in the market and others have been rumored to be ready for release. Unfortunately however, the effectiveness of such systems is not known since the techniques used in these products are often confidential and very little research about their effectiveness is available in the public domain.

In the last few years Gaines, Lisowski, Press and Shapiro [2], Umphress and Williams [7], Garcia [3], Leggett, Williams and Umphress [6], Leggett and Williams [5] and Young and Hammon [8] have studied the use of keystroke characteristics in verifying identity of a person. Gaines et al. [2] describe an experiment in which seven professional secretaries at the Rand Corporation were asked to type the same three passages of text at two different times separated by four months. All secretaries were not available to type all three texts at both sessions and complete data was available for only 11 sessions. Each of the three passages of lowercase text was about 300–400 words long. The first passage was an ordinary English text, the second a collection of random words while the third was a collection of random phrases. Keystroke latency times between adjacent letters (called *digraph latency times*) were computed for each individual and were found to vary from 75 msecs to several seconds. Also, it was found that there was little difference in the digraph times in the three passages and therefore, for each individual, the information from the three texts was merged. Since the digraphs considered involved only lower-case letters and spaces, there were 27×27 possible different digraphs. Most of these 729 digraphs either did not occur in the typed material or occurred only infrequently. The analysis therefore was based on only those digraph values that had at least 10 or more replications for each sitting of each individual. There were 87 such digraphs. These digraph values were transformed by removing the outliers and then taking the logs of the remaining values. Logarithms of the values were used because it was assumed that the raw data was log-normally distributed and the transformed data was found to be approximately normally distributed. A classical two-sample t-test of the hypothesis that the means of each digraph times at both sessions were the same was carried out assuming that the two variances were the same for each individual. It was shown that the number of digraph values that passed the test were typically between 80 percent to 95 percent.

Gaines et al. also studied the suitability of such digraph latency information in authenticating identity. As noted earlier, there were 87 digraph values that had 10 or more samples for each of the eleven different sessions. Each authentication test involved selecting one of these sessions as the reference session and each of the remaining 10 sessions as a session from a person (or claimant) wishing to access the computer system. A total of 55 such tests can be carried out given the data and the symmetry of the tests. Using the same t-tests, it was found that out of the 55 tests, the imposter[1] pass rate (percentage of invalid user attempts being accepted) was zero and the false alarm rate (percentage of valid user attempts being denied access) was about 4 percent (2 out of 55). Further analysis was carried out in an attempt to identify what Gaines et al. [2] call *key* or *core* digraphs. It was found that if only five digraph values (viz., in, io, no, on, and ul) were used, the authentication procedure worked perfectly i.e., no imposter pass or false alarms were found.

Although the results of [2] are encouraging, their study had a number of limitations. The most important being the number of individuals involved in the experiment. Their results therefore, particularly those relating to use of only five digraphs in authentication, need much further investigation.

Umphress and Williams [7], Leggett, Williams and Umphress [6] and Leggett and Williams [5] report the results of two experiments similar to the experiment conducted by Gaines et al. [2]. The first experiment had 17 programmers of varying typing ability provide two typing samples, the first with about 1400 characters that served as a reference profile and the second of about 300 characters that served as the test profile. In the second experiment, 36 participants typed in a 537 character passage at two different times separated by over a month. The basis of the research of [6] is to use two keystroke characteristics of the user. The first measure is the mean of the keystroke latencies of the user, essentially the user's typing speed. The second indicator involves comparing digraph latencies between all digraph combinations that have been typed by the user with reference latencies in a 26×26 reference latencies matrix whose rows correspond to the first letter of a two letter digraph and columns correspond to the second letter. In the second experiment, blank was added as a valid character in digraphs and the first part of the test that was based on the mean of all keystroke latencies of the user was dropped since it was found not to add any discriminating power to the verifier. Since many of the digraph latencies occur only infrequently, the standard deviation of the reference profile latencies (i.e., all the latencies in the reference profile) is used as a measure of tolerance of a match. If the test digraph latency time was within 0.5 standard deviations of the reference digraph latency mean then the latency was counted as valid. The ratio of valid digraph latencies to total latencies in the test string was then computed. If the ratio was above 0.6, the user was considered to have passed the verification test.

In [5] 12 different digraph latency tests were evaluated. These included using different maximum digraph latencies allowed to remove outliers (viz., 300 msecs, 500 msecs and 750 msecs) as well as applying the test to only a subset of the digraphs, for example, the subset identified by [2] or 6 and 15 most frequent digraphs,

[1] We are inclined to agree with the editor that "imposter" is better spelled as "impostor" but we will continue to use "imposter" since earlier papers in this field have used this spelling.

left-hand-only digraphs, right-hand-only digraphs, etc. It was found that if the five digraphs identified by [2] as core digraphs were used, the false alarm rate was above 30 percent and the imposter pass rate above 17 percent. The best results were obtained by using all digraphs involving lower-case letters only and the blank with a maximum latency of 500 msecs. This digraph latency test resulted in an identity verifier with false alarm rate of only about 5.5 percent and imposter pass rate of about 5.0 percent. Although the above low error rates are quite impressive, the imposter pass rate of 5 percent is still too high to be useful as an identity verifier since an imposter pass is a breach of the system security. A false alarm rate of 5 percent could well be acceptable since it would be nothing more than a nuisance in that a genuine user would, on the average, fail to get access to the system 1 out of 20 attempts. Therefore a reliable identity verifier would require techniques that would reduce the imposter pass rate to well below 1 percent. A lower false alarm rate would also be desirable but, as noted above, not essential.

The experiments discussed above have a major limitation in that they required the users to type in rather large character strings, first for generating the reference latencies data and then for verification. In spite of this, in the experiments of [6] it was necessary to use standard deviation of the reference profile latencies as a measure of tolerance of a digraph latency for each digraph latency in the matrix. Verification itself required the user to type in a large number of characters. For example, in the experiment of [2], a total of more than 1000 words needed to be typed by each claimant. The experiment of [7] required the user to type 300 characters while the second experiment reported in [5] required 537 characters. A static identity authentication system would not be successful if it asked the user to type long strings for reference purposes and another long string every time for verification purposes.

Garcia [3] describes a U.S. patent for a method and apparatus for identity verification based on a somewhat different approach. He suggests that the best data for identity verification is derived when an individual types his/her own name since the latencies generated by the user in typing his/her name have been found to be stable and unique. In addition, the name is the easiest password to remember. The first step of the procedure suggested by [3] involves the user typing his/her name a number of times to provide a vector of mean latencies to be used as a reference. This Garcia calls the *electronic signature* of the individual. In addition, the covariance matrix of the vectors of reference latencies is computed as a measure of the consistency of the individual's signature. In computing the vector of mean latencies and the covariance matrix, the outliers are removed.

When a person wants to access a computer resource, he is required to identify himself by typing in his/her name. The latency vector of the keystrokes of this name is compared with the reference signature that is stored in the computer. If this claimant's latency vector and the reference signature are statistically similar, the user is granted access to the system. The Mahalanobis distance function is used to measure the similarity of the two vectors. It is recommended in [3] that if the computed distance measure is more than 100, the vectors should be considered dissimilar and if less than 50, the vectors should be considered similar. If a value of between 50 and 100 is obtained, it is suggested that the claimant be required to retype the name. Although no evidence is presented, the suggested procedure is claimed to have an imposter pass rate of 0.01 percent and a false alarm rate of 50 percent. Garcia notes that the thresholds for acceptance and rejection may be altered if one wishes to reduce the false alarm rate and is willing to accept a higher imposter pass rate.

Garcia [3] also suggests another procedure which he calls *complex discrimination*. Rather than using the same string for an individual, complex discrimination involves the individual to type in each of at least 1,000 of the most common words in the English language at least 10 times to provide the reference raw data. Now, for verification, a random phrase is generated by the computer using the common words used in the reference and the user is required to retype that phrase. The latencies recorded by the user are then compared with the latencies stored in the computer and the user is permitted access only if similarity between the latencies is established. No information about the effectiveness of this approach is presented. The approach is of course not practical in most applications since it requires quite a long session to generate the reference data.

More recently, another U.S. patent for identity verification has been granted to Young and Hammon [8]. Young and Hammon use the term *keystroke dynamics* to denote the typing pattern of an individual including features like latencies and keystroke pressures. Although the details of the procedure used in the invention are not described clearly, it is suggested that a plurality of features be used. These could include digraph latency times, time to type in a predefined number of keystrokes, time to enter some common words (e.g., the, and, for). The identity verifier itself is based on first obtaining reference features (it appears that the features include keystroke latencies and possibly keystroke pressures) about a user and then comparing the vector of these features with similar features extracted from a claimant's typing session. The comparison is based on computing the Euclidean distance between the two vectors of features. No information is however presented about the effectiveness of the techniques used. Young and Hammon also propose that the identity verifier should typically operate on a continuous basis (this is sometimes called *dynamic* verification) in contrast to *static* verification that takes place only once at the start of each login session. Further, it is suggested that the keystroke timing device could be located in the terminal itself and the terminal could send the encoded timing information to the computer that the claimant wishes to access.

In this article, we use an approach for static identity verification similar to that used in [3]. Our approach is based on using keystroke information obtained during the login process using a modified login sequence. In addition to using the login name and password, we propose that the user be required to type in two additional strings that are familiar to the user, for example, the user's first and last names. An identity verifier using the latency information obtained when only user-name and password are used in the login process was found to provide good performance (around 1 percent imposter pass rate), but the additional two strings improved the performance considerably. Although more than four strings may be desirable to obtain accurate information about a user's keyboard characteristics, we feel a user cannot be expected to type in much more than four strings for verification purposes. Also we believe the information obtained from typing four well-known strings is likely to be more reliable than information obtained from a user typing in a large number of unfamiliar strings since typing familiar strings is less error prone and does not involve difficulties like reading text from paper and therefore provides a more distinct signature. Using the names as additional strings to be typed would provide data obtained from text strings that, for most people, would not change for their life time. Using the name could well be suitable for applications like the ATMs where the user could be asked not only to type in his/her PIN but also his/her name. The results of present study suggest that the combination of the PIN and the keystroke characteristics obtained when the user types in his/her PIN and his/her name could provide a very secure system.

IDENTITY VERIFIER

The proposed identity verifier uses the following approach. First, for each user, the procedure described below is followed to obtain a reference signature (analogous to a written sample signature) consisting of latency information recorded during the modified login process suggested earlier. Each time the user desires access to the computer system, he/she provides a digital signature (called the *test signature* or the *claimant's signature*) during the login process. The claimant's identity is verified if this digital signature matches the reference signature stored in the system.

To obtain a reference signature, we follow an approach similar to that used by the banks and other financial institutions. A new user goes through a session where he/she provides a number of digital signatures by typing in the four strings several times. Note that in the present environment the digital signature has four components, one component for each string that the user types. The system requires a new user to provide eight reference signatures by typing his/her username, password, first name and last name eight times. The number 8 was chosen to provide sufficient data to obtain an accurate estimation of the user's mean digital signature as well as information about the variability of his/her signatures. We discuss the impact of selecting fewer reference signatures in the upcoming section, "Evaluating the Verifier."

A signature or a component of a signature can be visualized by plotting characters typed versus latency times between successive keystrokes. The points thus obtained may be joined to obtain a signature "curve." In the present system the signature curve has four component curves (one corresponding to each of login name, password, first name, last name). A sample curve for a user with last name "Stephenson" is shown in Figure 1.

As illustrated in Figure 1, the first latency time stored is the elapsed time from the keystroke of the first letter to the second. The last latency time shown is the time from the last character to the carriage return. The latency from the time of the prompt to the striking of the first character is discarded as its variance can be high.

As noted earlier, the identity verifier would need to compare a test signature provided by the user wishing to access a computer system with the reference signature; allowing access if the test signature is *similar* to the reference signature. To carry out the comparison, a mean reference signature is first computed by calculating the mean and standard deviation of the 8 values for each latency. For each latency, the mean is then compared with each of the 8 values of that latency and any outliers (datum greater than three standard deviations above the mean) are discarded. This resulted in 0.85 percent of the latencies being discarded, the discards being distributed nearly uniformly over the string. The mean of the remaining values for each latency is now calculated. This process is repeated for each latency of the four login strings to produce four sets of mean latency values for each user. These four sequences (or "curves") are collectively referred to as the *mean reference signature* for the user.

Some of the reference signatures that had a latency discarded were studied to decide if all latencies from such signatures should be discarded and an additional reference signature requested. We found that the discarded latencies were isolated instances of large variances and the strings containing them usually had acceptable values of other latencies.

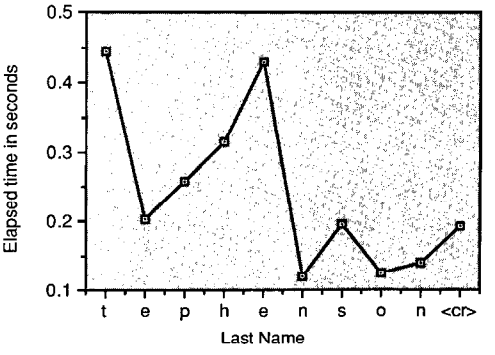

FIGURE 1. Latency Signature for "Stephenson"

A suitable technique is now needed for comparing a test signature with the mean reference signature. The approach used by [7] involves comparing individual latencies of the test signature with mean reference latencies and accepting the test signature as having been verified if more than 60 percent of the test latencies are valid. As discussed earlier, a latency was considered valid if it was within 0.5 profile standard deviations of the mean reference digraph latency. This approach has a major weakness in that some of the latencies in the two signatures being compared could differ substantially but the test signature could still pass the verification test.

Empirical investigations were carried out to evaluate the effectiveness of the approach used by [7] in our environment. The parameter values 0.5 and 0.6 used in the test were varied and results studied. Preliminary results suggest that the approach is not particularly reliable for comparing signatures.

Another approach of comparing two signatures is to look at each signature as a vector that consists of the set of 4 vectors of latency values. The mean reference signature, M, is then given by:

$$M = \{M_{\text{username}}, M_{\text{password}}, M_{\text{firstname}}, M_{\text{lastname}}\}$$

Now comparing M with a test signature, T, involves comparing the two vectors and determining the magnitude of the difference between them. In the ensuing discussion, let $M = (m_1, m_2, \ldots, m_n)$ and $T = (t_1, t_2, \ldots, t_n)$ where n is the total number of latencies in the signature.

The present verifier computes the magnitude of the difference between M and T as the l_1 norm:

$$\|M - T\|_1$$

given by:

$$\sum_{i=1}^{i=n} |m_i - t_i|$$

Although this approach works very well it has a weakness in that it does not take into account the shape of the signature curves. The difference between the shapes of the test and reference signatures could be significant even if the differences in the latency values are small. A more reliable comparison of M and T would probably include some technique of comparing the shapes of the signatures. We discuss some preliminary work on using the slopes of the lines between successive latencies as a measure of shape later in this article.

Once the magnitude of the difference between a given T and M has been computed, a suitable threshold for an acceptable size of the magnitude is required. We have chosen to set the threshold for each user based on a measure of the variability of his/her signatures. A user that has little variability in his/her signatures would have a small threshold while another user with large variability should have larger threshold for accepting his/her test signatures. We therefore need to compute a measure of variation between the 8 reference signatures, and the mean reference signature obtained from them as described above. Let the 8 training signatures be, $S_1, S_2, \ldots, S_n$. We calculate $\| M - S_i \|_1$ for $i = 1$ to 8. The mean and standard deviation of these norms are used to decide a threshold for an acceptable norm value of the latency difference vector between a given T and M. If we set the threshold value to be mean plus one standard deviation, we would expect the user to successfully login about 84 percent of the time (a false alarm rate of about 16 percent) assuming that the latency differences between the 8 reference signatures and the mean signature are normally distributed. A threshold value based on two standard deviations should provide a false alarm rate of less than 3 percent although the imposter pass rate with a larger threshold would obviously be expected to be larger. The threshold is presently defined as the mean plus one-and-one-half standard deviations.

The verification algorithm now works as follows. The claimant attempts a login thereby providing a test signature, T, to the system. The norm $\| M - T \|_1$ is computed and if this norm is less than the threshold for the user, the attempt is accepted, otherwise it is flagged as an imposter attempt. Figure 2 shows the four possible judgments of the verifier.

EVALUATING THE VERIFIER

The verifier was implemented on a SUN® 350 workstation. Thirty-three users with typing speeds, measured when login name/password and name was typed, varying from 14 to 111 wpm participated in the following trials of the authentication algorithm:

(1) Each user provided his/her reference signature by typing in their login name, password, first name and last name eight times.
(2) Once the reference signature was obtained, each user attempted to log on to his/her own account five times, yielding 165 total self login attempts. The target user data, both reference and self login attempts, were collected during a single session.
(3) Six of the above users were randomly selected as targets for the remaining 27 users. Each of the 27 users were allowed five imposter attempts for each of the six target users, yielding 810 imposter login attempts. All of the login information, including passwords, was given to each imposter but the imposters did not witness the target users' trials.

Not all users participating in the above trials knew about the purpose of the trials. One-half of the users were given no information about the verifier or the purpose of the trials. The remaining users were told what the trials were for and how the verifier worked, but were asked to login normally; i.e., they were asked not to try to exploit the verifier by employing unnatu-

SUN is a registered trademark of SUN Microsystems.

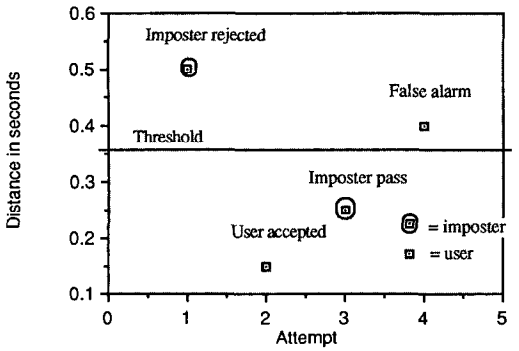

FIGURE 2. Possible Judgments of the Verifier

ral timing quirks that would make their signature extremely difficult to match.

The above 975 trials resulted in an imposter pass rate of 0.25 percent (2 out of 810) and a false alarm rate of 16.36 percent (27 out of 165) over all the trials. How the false alarm rate can be significantly reduced will be discussed later in this article.

We now present some results of the trials in detail. Figures 3–8 show results of six users (who were the targets of imposter attempts) logging in as themselves as well as attempts of other users logging in as imposters. Although each of the six users was the target of 27 imposters, we present results of only the eight closest imposters to keep the figures legible. The vertical axes in the graphs indicate $\| M - T \|_1$ in seconds. The horizontal axis shows the successive login attempts. Each line plotted depicts the 5 successive login attempts by one of the eight imposters. Each of the figures shows the target user typing rate, the number of characters in the signature, the mean and standard deviation of $\| S_i - M \|_1$, the authentication threshold, the best imposter attempt, the worst imposter attempt, the mean imposter attempt, and the number of false alarms and imposter passes.

Figures 3–8 summarize results of 840 trials including 30 reference signatures and 810 imposter attempts, with a total of 5 false alarms and 2 imposter passes. This leads to a false alarm rate of 16.67 percent (5 out of 30) and an imposter pass rate of 0.25 percent (2 out of 810). The false alarm rate is high, although it amounts, on the average, to only 1 out of 6 attempts being rejected (and therefore requiring another attempt). This rate can however be reduced. The threshold used in the above trials was the mean plus 1.5 standard deviations. As noted earlier, if the threshold is increased, the imposter pass rate should increase and the false alarm rate should decrease. We have studied the impact of varying the threshold on the false alarm rate and the imposter pass rate, as shown in Figure 9. These results show that the false alarm rate could be reduced substantially without a significant increase in the imposter pass rate if the threshold for verification was increased from 1.5 standard deviations to 2. The imposter pass rate at two-and-one-half standard deviations was still under 1 percent (7 out of 810), while the false alarm rate fell to 6.67 percent (2 out of 30).

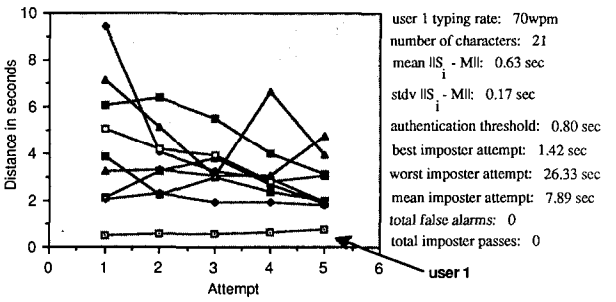

FIGURE 3. Login Attempts Against User 1

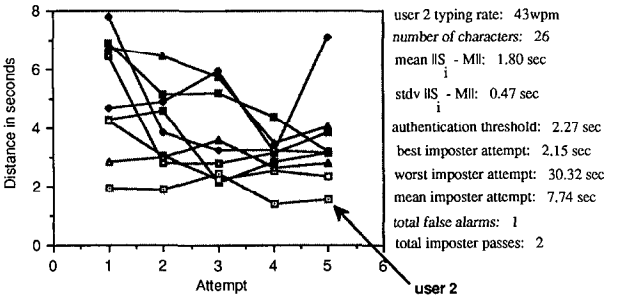

FIGURE 4. Login Attempts Against User 2

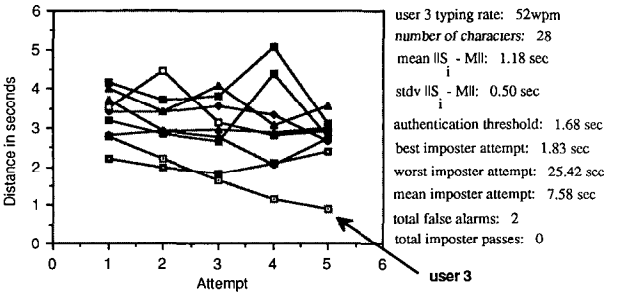

FIGURE 5. Login Attempts Against User 3

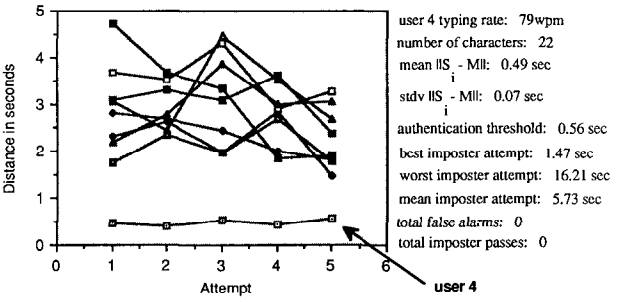

FIGURE 6. Login Attempts Against User 4

A detailed study of the imposter passes and false alarms was carried out. This has led to the following observations:

(1) Only two imposters were able to successfully pass as another user. The imposter passes involved a target user (user2) that had the highest authentication

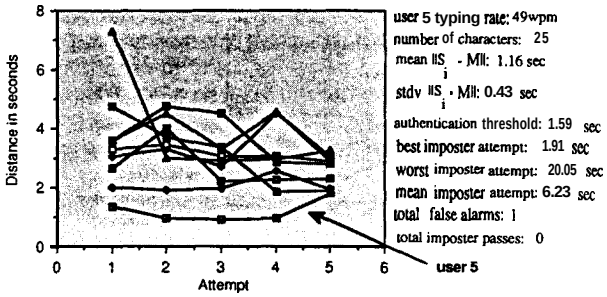

FIGURE 7. **Login** Attempts Against User 5

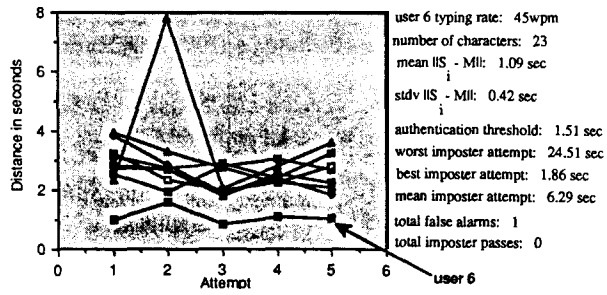

FIGURE 8. **Login** Attempts Against User 6

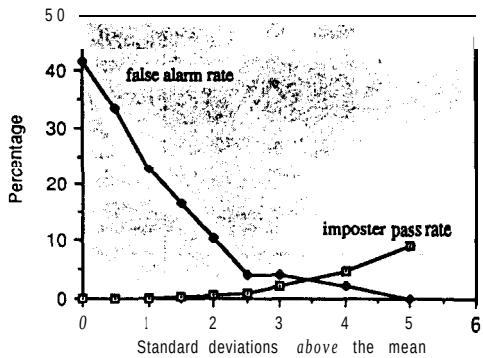

FIGURE 9. **IPR** and FAR versus Threshold

threshold (2.27 seconds as compared to 0.80, 1.68, 0.56, 1.59, and 1.51 seconds for the other five users in the group user1 to user6), and in Figure 9, accounts for one-half of the imposter passes at 3 standard deviations, and one-third of the imposter passes at 5 standard deviations. Since the verification threshold was chosen to be one-and-one-half standard deviations away from the mean, a high variation gives imposters an easier target. This is similar to the problems faced by financial institutions when some elderly people and people with physical disabilities are unable to supply a precise signature. In such cases, the verification scheme suggested in this article may be inappropriate and some other means of identity verification could well be required.

(2) There was no significant correlation between knowledge of the verifier and the ability of an imposter to match the reference signature of another user.

(3) The variation of imposter pass rate and false alarm rate with the number of reference signatures obtained is given in Figure 10. The use of eight reference signatures is supported although the experiments suggest that as few as six reference signatures might be sufficient. Since the reference signatures are signature samples that are used to estimate mean and standard deviations of the signature latencies, to obtain good estimates any fewer than six reference signatures could not be recommended.

(4) The experiments support the use of four strings in the signature. If only two strings were used (that is, login name and password), the imposter pass rate was found to be somewhat higher although the false alarm rate at one standard deviation threshold was about the same. Figure 11 shows two- and four-string login imposter pass rates for thresholds varying from the mean to the mean plus 5 standard deviations.

(5) During testing a number of users challenged the authors that they could successfully pass as another user. A number of such users were allowed to login as some other user. These imposter attempts were not organized and therefore detailed results are not presented but it is satisfying to note that all such attempts failed except one imposter who after observing the training session of the target, was able to pass as that user once in 57 attempts.

DISCUSSION AND FURTHER AREAS OF RESEARCH

An earlier version of this experiment was carried out using 600 imposter login attempts against six users [4]. Although the analysis was not as comprehensive as the results presented here, there are a few points worth noting:

- The results support those presented here, with a false alarm rate of 13.3 percent (4 out of 30), and an imposter pass rate of 0.17 percent (1 out of 600).
- Users with "easy-to-type" login sequences were easier targets for imposters than those with more complex typing patterns. Although "easy to type" is difficult to define, short signatures are generally easier targets for imposters. The target of the successful imposter attempt had only 16 characters in the four strings. Imposter login attempts with this user as the target showed the fastest mean typing rate of all imposter attempts, implying that users found the strings relatively easy to type. This is analogous to having a very simple signature that is relatively easy to forge. The difficulty arising from short signatures may be overcome in part by insisting on a minimum total length, and/or a minimum length for each of the four components of the signature.
- A zero imposter pass rate threshold exists for every target user in both experiments such that the false alarm rate is less than 40 percent (and typically is much lower). This means that there were no ob-

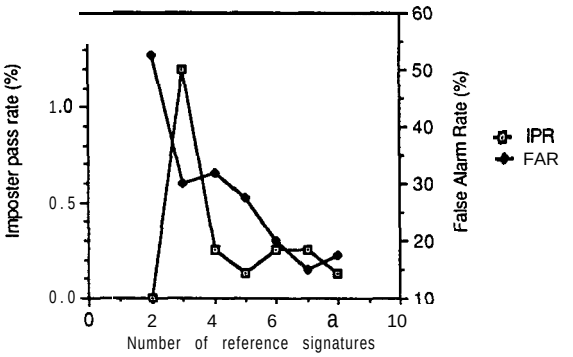

FIGURE 10. Impact of Number of Reference Signatures

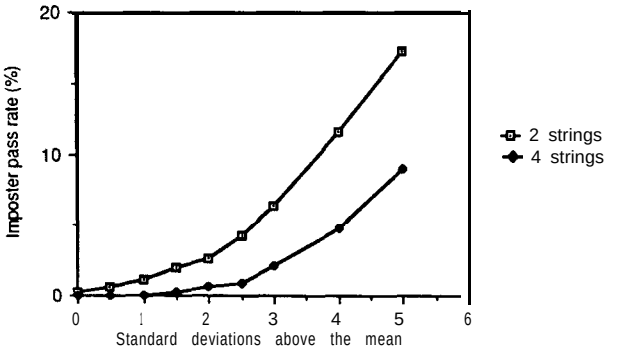

FIGURE 11. IPR versus Threshold for 2 and 4 String Login

served cases where an imposter did better than the target's best self-login attempt.

We now suggest a number of areas of further research. Firstly, a more comprehensive testing of the verifier is needed. Our testing included uses of varying computer literacy and typing abilities, but these were all between the ages of 20–45 years, and all were university students or staff. The "strongest" (i.e., most difficult to replicate) signatures appear to come from the most experienced computer users. A more comprehensive testing should include a wider cross section of users. Secondly, other measures of comparing M and T, and combinations of such measures need to be evaluated.

Improving the Performance of the Verifier

Although the present verifier is highly reliable with an imposter pass rate of less than 1 percent, further research is needed to reduce this rate.

As discussed earlier, a possible approach to improving the performance of the verifier would be to extend the comparison of the test signature and the mean reference signature to include a comparison of the shapes of the signatures. Handwritten signatures often have outstanding characteristics such as large loops or straight lines that serve as focal points during the identification process. To capture analogous information in a digital signature, we have looked at alternative measures that take this into account. Digital signatures often show sharp changes between successive latencies as a result of an individual's unique typing pattern.

Some preliminary testing has been done on a measure which measures the difference between the values of successive differences in latencies (which could be called "slopes" of the signature curves) in the test signature and the corresponding differences in the mean reference signature. Since we wish to highlight the outstanding differences in slopes as the distinctive features of a digital signature, the differences in the slopes were weighted by the amount of slope change in the reference signature. Let the vector of slopes of the mean reference signature M be denoted as I given by: $I = (i_1, i_2 \ldots i_{n-1})$, and similarly define J as the vector of slopes of the test signature T as: $J = (j_1, j_2 \ldots, j_{n-1})$, see Figure 12.

A measure of the difference between the shapes of M and T is given by:

$$\sum_{k=1}^{n-2} (|\, i_k - j_k \,| + |\, i_{k+1} - j_{k+1} \,|)w_k$$

where w_k is defined as:

$$w_k = \frac{|\, i_{k+1} - i_k}{\max|\, i_{k+1} - i_k \,|}$$

Figure 13 shows the imposter pass rate versus the false alarm rate for this measure over 100 imposter login attempts. Although the results are not quite as good as the total distance measure, we believe it is worthwhile to study the use of the combination of the two tests. Depending on the degree of correlation of the two measures, the combined test may bring the imposter pass rate to below 0.01 percent without increasing the false alarm rate significantly.

Knowledge of how the verifier works may be exploited to provide additional security. A user may wish to add unusual timing characteristics to his signature making it very difficult to forge.

Timing Accuracy

For the verifier to work, it is necessary to obtain accurate timing information with sufficient resolution. On a dedicated machine this may not be a problem, however on a time-sharing system where access may be through a variety of networks and hard/firm/software, sufficiently accurate timing information may be difficult to obtain. The best solution to this, as suggested by [8] would probably be for the keyboard to capture the latencies and transmit this information upon request. Further work is needed to address this problem.

Other Uses of the Verifier

Although we have only discussed identity verification aspect of the present work, it is likely that the techniques described here would prove useful in detecting whether a user is under the influence of alcohol or other drugs or is excessively tired. It is expected that the signature of a user would change, possibly substantially, under the influence of drugs although we have not had an opportunity to test this assumption.

A mechanism for detecting an intoxicated or tired

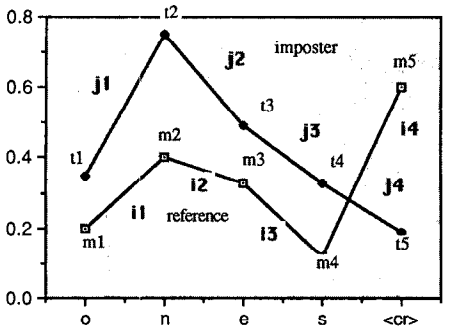

FIGURE 12. Using Slopes to Measure Distance

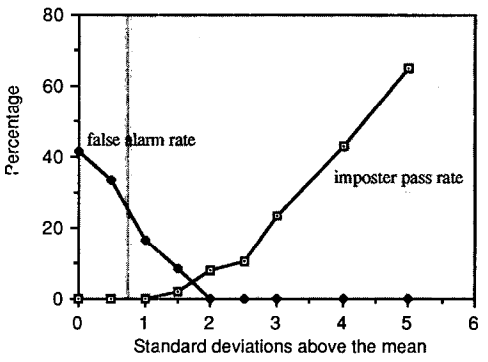

FIGURE 13. IPR and FAR versus Threshold for Slope Measure

user may be of use in security or safety sensitive installations where it may be important that the user (or operator) of the computer system be alert to deal with any emergencies that may arise.

CONCLUSIONS

We have described an identity verifier that uses keyboard latency information captured during a user's login process to verify the identity of the user. The verifier described was found to have an imposter pass rate of less than one percent when the imposter knew the target user's login name, first and last name, as well as the password.

We have also reported preliminary results of using a technique based on comparing signature shapes for identity verification. These results are encouraging, suggesting the possibility of combining the two techniques proposed in this article to further reduce the imposter pass rate.

Acknowledgments. The authors wish to thank Cameron Gregory and Jiang Yi for their efforts in collecting the data, as well as all the users who happily participated in our experiments. We also wish to thank the referees, one of whom made a number of interesting comments.

REFERENCES

1. Card, S.K., Moran, T.P., and Newell, A. The keystroke-level model for user performance time with interactive systems. *Commun. ACM 23*, 7 (July 1980), 396–409.
2. Gaines, R., Lisowski, W., Press, S., and Shapiro, N. Authentication by keystroke timing: Some preliminary results. Rand Report R-256-NSF. Rand Corporation, Santa Monica, CA, 1980.
3. Garcia, J. Personal identification apparatus. Patent Number 4,621,334. U.S. Patent and Trademark Office, Washington, D.C., 1986.
4. Joyce, R., and Gupta, G. User authentication based on keystroke latencies. Technical Report #5, Department of Computer Science, James Cook University, Australia, 1989.
5. Leggett, J., and Williams, G. Verifying identity via keyboard characteristics. *Int. J. Man–Machine Studies 23*, 1 (Jan. 1988), 67–76.
6. Leggett, Williams, G., and Umphress, D. Verification of user identity via keyboard characteristics. In *Human Factors in Management Information Systems*, J.M. Carey, Ed., Ablex Publishing, Norwood, NJ.
7. Umphress, D., and Williams, G. Identity verification through keyboard characteristics. *Int. J. Man–Machine Studies 23*, 3 (Sept. 1985), 263–273.
8. Young, J.R., and Hammon, R.W. Method and apparatus for verifying an individual's identity. Patent Number 4,805,222. U.S. Patent and Trademark Office, Washington, D.C., 1989.

CR Categories and Subject Descriptors: K.6.m [**Management of Computing and Information Systems**]: Miscellaneous—*security*
General Terms: Security
Additional Key Words and Phrases: Authentication, identity verification, identification, keystroke latencies

ABOUT THE AUTHORS:

RICK JOYCE is a member of the Technical Staff at AT&T Bell Laboratories. His current research interests include design of client/server based common graphics platform for network operations, and dynamic identity authentication. Author's Present Address: AT&T Bell Laboratories, 480 Red Hill Road, Middletown, NJ.

GOPAL GUPTA is a professor and Head of the Department of Computer Science at James Cook University. His current research interests include data structures and database management systems. Author's Present Address: James Cook University, Department of Computer Science, Townsville, Qld 4811, Australia.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.