*Int. J. Man-Machine Studies* (1985) **23**, 263–273

# Identity verification through keyboard characteristics

DAVID UMPHRESS AND GLEN WILLIAMS
*Department of Computer Science, Texas A & M University, College Station, Texas 77843-3131, U.S.A.*

(*Received 15 January 1985 and in revised form 25 April 1985*)

Most personal identity mechanisms in use today are artificial. They require specific actions on the part of the user, many of which are not "friendly". Ideally, a typist should be able to approach a computer terminal, begin typing, and be identified from keystroke characteristics. Individuals exhibit characteristic cognitive properties when interacting with the computer through a keyboard. By examining the properties of keying patterns, statistics can be compiled that uniquely describe the user. Initially, a reference profile is built to serve as a basis of comparison for future typing samples. The profile consists of the average time interval between keystrokes (mean keystroke latency) as well as a collection of the average times required to strike any two successive keys on the keyboard. Typing samples are scored against the reference profile and a score is calculated assessing the confidence that the same individual typed both the sample and the reference profile. This mechanism has the capability of providing identity surveillance throughout the entire time at the keyboard.

## Introduction

With increasing frequency, newspaper headlines are emphasizing the vulnerability of computer security measures. Numerous stories detail the break-in of computer systems, the perusal of classified files, and the destruction of invaluable information. Reading such stories should give the computer-professional pause to consider how system resources are protected. In examining methods by which users gain access to computer resources, the first items that come to mind are passwords, keys, badges, fingerprints and signatures. Even at this superficial level, it is not difficult to see that security mechanisms can be grouped into three general categories (Wood, 1978):

(1) what the user knows, e.g. passwords;
(2) what the user has, e.g. keys, badges;
(3) what the user is, e.g. fingerprints, signatures.

The degree of confidence with which each category protects computer resources lies in how readily the security mechanism can be circumvented. Each category above has advantages in terms of cost, ease of implementation, and user convenience. However, it is the third category, "what a user is", that presents the strongest line of defence against counterfeit users.

Close examination of most computer security systems reveals that they are not personal identifiers, but identity verifiers; that is, they do not require a multitude of identity measurements to produce a composite confidence of user identity. Instead, they require a single source of identity and verify authorization based on that source alone. The password is by far the most popular identity verifier due to its economical viability and ease of implementation: it requires no special hardware. Further, it is



0020-7373/85/090263+11$03.00/0

intangible, and therefore easy to transport from terminal to terminal and physically difficult to lose. However, the very intangibility of the password gives it the distinct disadvantage of being compromised without consent or knowledge of its disclosure. Moreover, it does not provide protection past the initial recognition stage.

The widespread use of the personal computer further compounds the problem of sole-source identity verification. When furnished with even the simplest of telecommunications equipment, the user of the home computer may initiate an automated search for computer dial-up ports. Once a dial-up port has been found, exhaustive attempts may be made to determine a valid password and thus allow the user to impersonate an authorized user. After circumventing the initial security measures, the imposter is free to browse files and scavenge information (Steinauer, 1981). In essence, the wide proliferation of the personal computer has permitted abusers to pick the lock of many data-processing installations.

The havoc unleashed by the home computer can be turned to an advantage. This paper examines a means of supplementing identity verification using the equipment that is available on existing personal computers. Further, the concept described not only adds to initial identity verification, but provides a means of constant identity surveillance.

## Predictable human characteristics

As early as the turn of the century, psychology experiments demonstrated that the mechanics of human actions are predictable in the performance of repetitive, routine tasks. In 1895, observation of telegraph operators showed that each operator had a distinctive pattern of keying messages over telegraph lines (Bryan & Harter, 1973). Moreover, operators often recognized who was transmitting information simply by listening to the characteristic pattern of dots and dashes.

Just as the telegraph key served as a common input medium in days past, keyboards, light pens, joysticks and mice are common input devices today. The question posed now is one of whether properties exhibited in the use of these devices are unique to the individual user. Keyboard characteristics are rich in cognitive qualities and give great promise as a personal identifier. Anyone who sits within earshot of a typist or has an office next to a keypunch room is usually able to recognize typists by keystroke patterns. This paper presents a study of keystroke patterns as a supplement to identity verification. Part 1 establishes the framework for using keystroke characteristics as an identifier. Part 2 describes the results of an actual experiment in personal recognition.

## 1. Model of human behaviour

Human nature dictates that an individual does not sit before a computer and deluge the keyboard with a furious and continuous stream of nonstop data. Instead, the user types for a while, pauses to collect thoughts, types a bit more, pauses again to decide a new strategy, continues typing, and so forth. In developing a scheme for identity verification, a common base must be established for determining which keystrokes characterize the individual's key patterns and which do not. Psychological models describing the human interface with computer programs aid in this process. Several models are proposed in the literature. The most popular, the keystroke-level model,

was chosen as a basis for this work (Card, Moran & Newell, 1980). It describes man-machine interaction during a session at a computer terminal. The model was intended as a tool for the evaluation and comparison of designs for highly interactive programs. Given this scope, however, it provides an interesting insight into human performance and, more importantly, human predictability.

The keystroke-level model summarizes the entire terminal session as:

$$T_{\text{task}} = T_{\text{acquire}} + T_{\text{execute}}$$

$T_{\text{task}}$ represents the duration of the terminal session; $T_{\text{acquire}}$ is the time required to assess the task, build a mental representation of the functions to be performed and choose a method for solving the problem and $T_{\text{execute}}$ is the time required to call on the system resources to perform the tasks.

As one would expect, $T_{\text{acquire}}$ varies according to the magnitude of the task at hand, the experience of the user and understanding of the functions to be performed. It is not quantifiable, therefore, and cannot be used to characterize individuals. $T_{\text{execute}}$, on the other hand, describes mechanical actions. It may be described further as:

$$T_{\text{execute}} = T_k + T_m \dagger$$

where $T_k$ is the time to key in information and $T_m$ is the time required for mental preparation. The $T_m$ here may be thought of as tactical planning in contrast to $T_{\text{acquire}}$ which is strategic in nature.

Such a macro view of $T_{\text{execute}}$ does not depict the true processes that are occurring. When interacting with a program, the user does not separate his or her actions into mental time followed by keystroke time. Instead, the two are intermixed. Looking closer, $T_{\text{execute}}$ can be portrayed as a series of mental/keystroke clusters as:

$$T_{\text{execute}} = \Sigma(T_{mi} + T_{ki}).$$

The expression in parentheses describes the fundamental human action of breaking a larger task into smaller, more easily managed, subtasks. Each subtask is known as a cognitive unit, or in more vernacular terms, a "chunk".

The representation of a data item being keyed into the computer can be seen with the following example. Suppose a user wishes to display the directory of a disk. Further, suppose that the command needed to accomplish this is, say DIR. The keystroke-level model would represent the actions required as:

MK[D]K[I]K[R]K[RETURN].

M constitutes $T_m$, the time required to conceptualize which keys must be activated in order to display the directory. The collection of Ks make up the actual keystrokes, the corresponding keystroke is enclosed in brackets.

Extrapolating from this, single commands can be linked serially to form entire typed lines. The authors of the model give elaborate heuristics for determining the placement of the M operator within the keyed input in an effort to show cognitive patterning. Intuitively, it is expected that those keystrokes within each cognitive unit are chosen as being characteristic of an individual's typing patterns. Examining keystroke patterns that span cognitive boundaries introduces the complicating factor of pauses caused by

† The keystroke-level model includes operators for time required to point a mouse, time required to draw lines, etc. These operators are independent of keystroke time and hence not included for discussion.

mental preparation time. This is a factor that is not necessarily quantifiable. Instead keystrokes between M operators are most representative of individual key patterns.

Close examination reveals that even at the granularity of the cognitive unit identified by the model, certain keystrokes must be filtered further. Research has shown that when a typist is keying data, the brain acts as a buffer. The typist first looks at the text to be typed, loads a certain amount of text into the buffer, then outputs the text onto the keys of the keyboard. The buffer is, on the average, 6–8 characters in length (Shaffer, 1973). Due to this limitation in buffer size, typists group symbols into smaller cognitive units and pause between each unit. Typical pause points are between words as well as within words that are longer than 6–8 characters (Cooper, 1983). In light of this, the definition of a predictable cognitive unit must be restricted. Only keystroke patterns within the first 6–8 characters of words will be considered as candidates for characterization.

## 2. Experiment in keystroke characterization

The psychological model describes which keystrokes provide meaningful information towards an individual's particular key pattern. The problem now becomes one of collecting statistics that characterize the key pattern. Two sets of inputs are required for user identification, a reference profile and a test profile.

### REFERENCE PROFILE

The reference profile serves as a control; it is the basis for all subsequent comparisons to determine personal identification. To build a reference profile, an individual takes a standard typing test. Each keystroke is time-tagged and stored for analysis. The participant is instructed not to attempt to correct typing errors. Typists often realize they have made a mistake immediately after the error has been made (Shaffer, 1970). In the process of catching the mistake, the typist unconsciously pauses before completing the remainder of the word. This introduces an extraneous cognitive boundary that misrepresents the normal keystroke pattern. For this reason, the first step in the filtering process is to compare the test keystrokes to the test text. Words containing errors are discarded. This is done in an effort to obtain as clean a reference profile as possible.

After screening the entered text for errors, the keystrokes are grouped into words. Keystroke latencies are calculated by taking the difference between the times of each pair of adjacent keystrokes. Latencies are calculated only for the first six keystrokes in each word. If the word is longer than six characters, the remaining keystrokes are ignored.

The final steps of the elimination process attempt to compensate for possible anomalous keystroke latencies. First, latencies over 0·75 s in duration are discarded. Latencies of over 0·75 s indicate that the typist is unfamiliar with the keyboard (Fig. 1). Thus, that particular keystroke is not a good candidate for inclusion in the typist's keystroke profile. Second, the latency time for capital letters is halved to allow for the two keystrokes required to form capital letters. Since latency times are considered only for keystrokes within words, a place where capital letters seldom appear, this has little effect on the pattern recognition process.

Two measures of key patterning are produced from the filtered keystrokes. The first measure is the mean and standard deviation keystroke latency. The second indicator

| Skill level | Keystroke speed (wpm) | Keystroke latency |
|---|---|---|
| Best | 135 | 0·08 |
| Good | 90 | 0·12 |
| Average skilled | 55 | 0·20 |
| Average non-skilled | 40 | 0·28 |
| Poor | 25 | 0·48 |
| Unfamiliar with keyboard | | 1·20 |

FIG. 1. Comparison of keystroke speeds (Card *et al.*, 1980).

describes the latency between all adjacent letter combinations by defining a $26 \times 26$ matrix, whose rows correspond to the first letter of a two letter digraph, and whose columns correspond to the second letter. Each cell in the matrix gives the average keystroke latency of the diagraph defined by the cell position. Fig. 2 summarizes the process used to form the reference profile.

## TEST PROFILE

The test profile is the collection of keystrokes produced by an individual requesting identity verification. The crux of the problem here is to compare the test profile with a reference profile and assign a confidence that the individual typing the test profile is the same as the one who typed the reference profile. Two methods of comparison are possible. One approach treats the test profile in much the same way as the reference profile was analysed. Here, keystrokes are collected, time-tagged and screened for errors. Spurious latencies are eliminated, statistics are computed and a matrix of digraph latencies is constructed. The statistics and matrix are then compared with the reference profile at one time in a batch-type manner. The second comparison method evaluates keystrokes in real-time.

Since an implementation strategy was devised based on real-time performance aspects, the second method will be discussed in depth. Using this method, keystrokes of the test profile are time-tagged in the same manner as in the reference profile. After each key is depressed, the difference between the time from the previous keystroke is calculated in real time. Keystrokes are filtered according to the cognitive principles discussed for the reference profile. First, only keystroke latencies within words are considered for comparison. That is, if any keystroke of the current digraph is other than an alphabetic character, the latency time is ignored. Second, if the second character is a backspace, an error is assumed to have occurred within the word. This and all subsequent keystroke latencies within the word are ignored. Third, only latencies between the first six characters of words are considered. Finally, latencies over 0·75 s in duration are discarded.

## SCORING

Two tests are performed to determine how closely the test profile matches the reference profile. The first test assesses keystroke intervals within character patterns and the second test appraises overall typing characteristics.

For the first test, the keystroke latency is compared to the appropriate cell in the digraph matrix of the reference profile. The cell position is determined by using the

| Time-tag keystrokes | | Screen errors | | Calculate latencies within words | | Eliminate latencies over 0·75 s | | Screen words over six characters |
|---|---|---|---|---|---|---|---|---|
| N0·0 | | N0·0 | | N | | N | | N |
| o0·28 | | o0·28 | | o0.28 | | o0.28 | | o0.28 |
| w0·47 | | w0·47 | | w0·12 | | w0·12 | | w0.12 |
| 0·67 | | 0·47 | | | | | | |
| i0·95 | | i0·95 | | i | | i | | i |
| s1·10 | | s1·0 | | s0·15 | | s0·15 | | s0·15 |
| 1·31 | | 1·31 | | | | | | |
| t1·52 | | | | | | | | |
| e1·63 | | | | | | | | |
| h1·81 | | | | | | | | |
| 1·97 | | 1·97 | | | | | | |
| t2·15 | | t2·15 | | t | | | | |
| i2·26 | | i2·26 | | i0·11 | | | | |
| m3·16 | | m3·16 | | m0·90 | | | | |
| e3·32 | | e3·32 | | e0·16 | | | | |
| 3·47 | | 3·47 | | | | | | |
| f3·65 | ⇒ | f3·65 | ⇒ | f | ⇒ | f | ⇒ | f |
| o3·79 | | o3·79 | | o0·14 | | o0·14 | | o0·14 |
| r3·97 | | r3·97 | | r0·18 | | r0·18 | | r0·18 |
| 4·16 | | 4·16 | | | | | | |
| a4·32 | | a4·32 | | a | | a | | a |
| l4·47 | | l4·47 | | l0·15 | | l0·15 | | l0·15 |
| l4·64 | | l4·64 | | l0·17 | | l0·17 | | l0·17 |
| 4·82 | | 4·82 | | | | | | |
| A5·05 | | A5·05 | | A | | A | | A |
| m5·32 | | m5·32 | | m0·27 | | m0·27 | | m0·27 |
| e5·55 | | e5·55 | | e0·23 | | e0·23 | | e0·23 |
| r6·02 | | r6·02 | | r0·47 | | r0·47 | | r0·47 |
| i6·22 | | i6·22 | | i0·20 | | i0·20 | | i0·20 |
| c6·49 | | c6·49 | | c0·27 | | c0·27 | | c0·27 |
| a6·77 | | a6·77 | | a0·28 | | a0·28 | | a0·28 |
| n7·09 | | n7·09 | | n0·32 | | n0·32 | | |
| s7·37 | | s7·37 | | s0·28 | | s0·28 | | |

Build matrix of digraph latencies and calculate stats

| | A | C | E | I | L | M | O | R | S | W |
|---|---|---|---|---|---|---|---|---|---|---|
| A | | | | | 0·15 | 0·27 | | | | |
| C | 0·28 | | | | | | | | | |
| E | | | | | | | | 0·47 | | |
| F | | | | | | | 0·14 | | | |
| I | | 0·27 | | | | | | | 0·15 | |
| L | | | | | 0·17 | | | | | |
| M | | | 0·23 | | | | | | | |
| N | | | | | | | 0·28 | | | |
| O | | | | | | | | 0·18 | | 0·12 |
| R | | | | 0·20 | | | | | | |

Mean latency: 0·22
Standard deviation: 0.09

previous character as the row identifier and the current character as the column identifier. The standard deviation of the reference profile is used as a measure of tolerance. If the test latency is within 0·5 S.D. of the corresponding latency in the reference digraph matrix, the test keystroke is considered valid.‡ A count is maintained of all intervals that pass this test as well as all intervals tested. At any point in time, the degree of pattern matching can be obtained by computing the ratio of valid intervals to total intervals.

For the second test, a count is kept of keystrokes that passed the filtering stage. A running sum is maintained on all these latencies as well. At any point in time, the mean latency and standard deviation may be calculated. The mean is compared to the reference profile mean using a standard two-tailed $t$-test for a population mean assuming a normal distribution. The null hypothesis is that the mean test keystroke latency is equal to the mean reference latency. 0·05 is used as the probability of a type I error, i.e. the chances of rejecting the null hypothesis when it is true. The formula to calculate the test statistic is given in Fig. 3. The null hypothesis will be rejected if

$$\text{abs}\,(z) > z_{0\cdot05/2},$$

where $z_{0\cdot05/2}$ is the area under a normal curve for a type I error of 0·05.

$$z = (\text{testmean} - \text{refmean})/(\text{teststd}/\text{sqr}(\text{testsize}))$$

where testmean: mean test latency
refmean: reference mean latency
teststd: test standard deviation
testsize: test sample size

Test fails if abs $(z) > z_{0\cdot05/2}$, where $z_{0\cdot05/2}$ is an alpha level of 0·05.

FIG. 3. Test statistic.

## SCORING RATIONALE

The digraph test gives a measure of whether a certain percentage of the letter combinations are within the tolerance of the reference profile. It was experimentally determined that, on the whole, the test digraph will differ from the corresponding reference digraph by approximately 0·5 of the overall reference standard deviation. Obviously, the test and reference values will not be exact. This is due not only to variation in individual typing speeds, but also by anticipation of letters beyond the most immediate diagraph. The classical example of anticipation is the typing of the words "whig" and "whim". The keyboard position of the last letter in each word affects the speed with which the first three letters are typed (Schaffer, 1982). In any case, the keystroke latency is normally within ±0·5 of the overall reference standard deviation. Further experimental results showed that when the reference profile and test profile are typed by the same individual, the ratio of valid intervals to total intervals is greater than 0·60.

Although the digraph test would appear to be sufficient to assess personal recognition, such is not the case. It does not weigh the final score in any way by the size of the test profile. The test for overall keystroke characteristics provides a simple means for

‡ The decision to use 0·5 standard deviation as a range of acceptability was based on experimental results. All profiles were scored using 2, 1, 0·5 and 0·25 S.D. as a measure of tolerance. The value of 0·5 gave the best overall score when comparing profiles typed by the same individual. It similarly produced the lowest score when comparing profiles typed by different individuals.

biasing the score based on the number of valid test keystrokes. An added bonus is that it generally verifies the results of the digraph test as well.

The combined results of the tests are used to designate the degree of confidence that the test typist is the same as the reference typist. The digraph test indicates a measure of character patterns while the overall keystroke test measures general keystroke characteristics. If a test profile passes both tests, a high confidence is assessed. Here, the test typist and reference typists are considered as the same individual. If one of the tests fails, a medium confidence is assigned. Finally, if both tests fail, a low confidence is assumed.

## Experimental results

Hardware for the experiment consisted of an IBM Personal Computer, a colour monitor, and a standard IBM PC keyboard. No modifications to the hardware were made. Three PASCAL programs constituted the software for the research, the first monitored keystroke inputs, each keystroke being time-tagged to the nearest 1/100 s and stored on a floppy disk. The second program analysed keystrokes and produced a database of reference profiles for each individual participating in the experiment. The profiles were created according to the filtering method described earlier. The third program compared test profile keystrokes to reference profiles and assigned scores based on the results of the comparison.

Seventeen people participated in the experiment. Participants ranged in proficiency from experienced touch-typists to those with no formal typing skills. All participants were experienced programmers. Each person was asked to take two typing tests. The tests were, for the most part, separated over several days. For the first test the individuals were asked to type approximately 1400 characters of prose. The keystrokes from this test were analysed and distilled into reference profiles. The second typing test, the test profile, consisted of 300 characters of prose.

| | | Test Profiles | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| R | 1 | H | | | | | | | M | | | | | | | | | |
| e | 2 | | H | | | | M | | | | | | | | | | | |
| f | 3 | | | M | | | | M | | | | | | | | | | |
| e | 4 | | | | H | M | | | | M | | | | | | | | |
| r | 5 | | | | M | H | | | | | | | | | M | | | |
| e | 6 | | | | | | H | | | | | | | | | | | |
| n | 7 | | | M | | | | H | | | | | | | | | | |
| c | 8 | M | | | | | | | H | | | | | | | | | |
| e | 9 | | | | M | | M | | | H | | | | | | | | |
| | 10 | | | | | | | | | | H | | | | | | | |
| P | 11 | | | | | | | | | | | H | | | | | M | |
| r | 12 | | | | | M | | | | | | | M | | | | | |
| o | 13 | | | | | | | | | | | | | H | | | | |
| f | 14 | | | | | | | | | M | | | | | H | | | |
| i | 15 | | | | | | | | | | | | | | | H | | |
| l | 16 | | | | | | | | | | | | | | | | H | |
| e | 17 | | | | M | | | | | | M | | | | | | | H |

FIG. 4. Comparison of reference to test profiles. H = high confidence; M = medium confidence; blank = low confidence.

Results of the comparison of all test profiles to all reference profiles are given in Fig. 4. The diagonal of the matrix in the figure shows a high degree of correlation when the same individual typed both reference and test profiles. Several medium confidence levels were assigned in instances where the typist of the profiles differed. For the most part, however, test profiles had low scores when the typist was not the same person who typed the reference profile.

Closer analysis of medium confidences assigned on the main diagonal showed that the individuals were not touch-typists, but typed with a high degree of variation. It was interesting to note that the medium confidence was assigned because the general characteristics test failed although the digraph test passed. In contrast, several medium confidences were assigned to profiles from different typists. In these cases the general test passed while the digraph test failed. From this it would appear that the digraph test alone may be an accurate measure of identity if the test sample size is large enough. Research is currently underway to determine the smallest number of keystrokes in the test profile that will satisfy recognition constraints.

## Evaluation

The success of this approach to identity verification can be defined in terms of several factors: false rejection rate, false acceptance rate, time to assess identity verification, convenience to the user and cost of recognition hardware (U.S. National Criminal Justice Information and Statistics Center, 1979).

The two most common gauges of security measures are false-rejection and false-acceptance rates. The false-rejection rate of a system gives an indication of how often an authorized individual will not be properly recognized. The false-acceptance rate describes how often an unauthorized individual will be mistakenly recognized and accepted by the system. The false-acceptance rate is generally more indicative of the level of security of a mechanism. This is due to the fact that it describes the degree to which the security measure may be breached by intruders. The false-rejection rate is important in that it describes the amount of user frustration in using the security technique. For purposes here, the false-rejection rate is obtained by examining the main diagonal of the matrix in Fig. 4. The diagonal describes the results of verification when both the reference and test profiles were typed by the same individual. The "worst-case" rate is computed as the ratio of low and medium confidence scores on the diagonal to the total number of scores on the diagonal. From Fig. 4, two of 17 individuals were not assessed with high-confidence scores. This gives a sample false-rejection rate of 12%. Similarly, the worst-case false-acceptance rate is determined by dividing the number of medium and high confidences outside the main diagonal by the total number of comparisons outside the diagonal. Of the 272 comparisons made, 16 were assessed with a medium confidence. This gives a false-acceptance rate of 6%.

Time required to assess identity verification is kept to a minimum by calculating keystroke latencies in real-time as keys are depressed. The major computing load is incurred when the confidence score is calculated. The number of operations is limited to five arithmetic operations and three comparisons. Thus, the time required to verify identity is minimal.

The main thrust of identification by the use of keystroke characteristics is user convenience. Identification is verified through user actions, i.e. the process of interacting with the computer via the keyboard. Convenience is maximized by virtue of the fact that surveillance is carried out while the user accomplishes a useful task. The user does not have to be cognizant of the fact that a protection mechanism is being used. Convenience is further heightened because this approach to security can be implemented using off-the-shelf personal computers.

From these factors evaluating the worth of using keystroke patterns as a means of identification, several points are evident. First, use of keystroke characteristics is not in itself sufficient for personal identification. Used in conjunction with another recognition technique, however, effective protection can be achieved. Secondly, in supplying data necessary for determining identification, the user can be performing a useful function. Finally, the experiment described involved users performing textual transcription. Collection and evaluation techniques may vary according to the task being performed. For instance, keying program source code may differ from entering commands to a text editor. If identity verification is to be performed on a continual basis, the nature of the task must be considered.

## Conclusion

Ideally, the user should be able to approach a terminal, begin typing and be identifiable from keystroke characteristics alone. No other form of identification would be necessary. This would remove the burden that is now placed on the user by current verification systems. Human predictability can be used as a means of personal identification. However, human variance places restrictions on the precision of recognition techniques. Keystroke patterns are rich with individual mannerisms and traits. The method discussed here described the rhythm and timing aspects of typing that can be used for identity verification. Although this research has not produced a means by which keystrokes alone can be used for identification, it has shown that, combined with other security techniques, keystroke patterns provide an effective means of secondary identification and a powerful means of identity surveillance.

## References

BRYAN, W. L. & HARTER, N. (1973). Studies in the physiology and psychology of the telegraphic language. *The Psychology of Skill: Three Studies.* Ed H. Gardner and Judith K. Gardner, Eds, pp. 35–44. New York: New York Time Co.

CARD, S. K., MORAN, T. P. & NEWELL, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, **23**, 396–410.

COOPER, W. E. (1983). *Cognitive Aspects of Skilled Typewriting*, pp. 29–32. New York: Springer-Verlag.

SHAFFER, L. H. (1970). The basis of transcription of skill. *Journal of Experimental Psychology*, **84**, 424–440.

SHAFFER, L. H. (1973). Latency mechanisms in transcription. *Attention and Performance*, Vol. IV, S. Kornblum, Ed. New York: Academic Press.

SHAFFER, L. H. (1982). Rhythm and timing in skill. *Psychological Review*, **89**, 116–117.

STEINAUER, D. (1981). Security in small computer systems. *Data Security Management.* Portfolio 84-04-03. Pennsauken, NJ: Auerbach Publishers, Inc.

U.S. NATIONAL CRIMINAL JUSTICE INFORMATION AND STATISTICS CENTER. (1979). *Criminal Justice Resource Manual: Computer Crime.* Washington, D.C.: Government Printing Office.

WOOD, H. M. (1978). The use of passwords for controlling the access to remote computer systems and services. *Computers and Security,* Vol. III. C. T. Dinardo, Ed., p. 137. Montvale, New Jersey: AFIPS Press.