CONTINUOUS AUTHENTICATION BY ANALYSIS OF KEYBOARD TYPING CHARACTERISTICS

S J Shepherd

Bradford University, United Kingdom.

ABSTRACT

This paper describes a simple, software based keyboard monitoring system for the IBM PC for the continuous analysis of the typing characteristics of the user for the purpose of continuous authentication. By exploiting the electrical characteristics of the PC keyboard interface together with modifications to the internal system timer, very accurate measurements can be made of keystroke interval and duration, including measurements of rollover. Rollover patterns, particularly when typing common diphthongs, can be highly characteristic of individual users and provide quite an accurate indication of the users identity.

KEYWORDS

Authentication, biometrics, typing cadence, PC.

INTRODUCTION

One of the major problems in the implementation of secure systems is the authentication of permitted users. The CCITT X.509 Authentication Framework within the OSI reference model is very much concerned with this subject, and is devoted to secure implementation methodology. But irrespective of the security service offered, each one ultimately depends upon reliable authentication.

The most popular authentication technique in current use is the *password*. This has the advantage of simplicity not only for the system designers but also for the end user, especially if they have little or no technical knowledge. The disadvantage of the password system, of course, is the ease with which the passwords themselves may be compromised, either deliberately by untrustworthy personnel, by accident, or by guessing [1]. Studies have shown that just over 400 passwords cover approaching 90% of the passwords in common use! The success of the *Internet worm* was largely due to it's internal library of such passwords which it applied repeatedly to guess its way into systems.

To gain stronger authentication, it is necessary to resort to techniques that are much more difficult, if not impossible, to forge. Previous studies [2] have identified three main categories of criteria that may be used to identify a user

- something the user *knows* e.g. a password or other secret fact;
- something the user *has* e.g. a smart card or token;
- something the user *is* e.g. a fingerprint or retinal pattern.

As stated, the first option is simple to implement but is limited in the level of security provided. The second option is better but requires the user to physically carry an item of hardware which can be lost or stolen. The strongest option by far is the last as it cannot be transferred between individuals in the same way as knowledge or hardware. However, the cost of equipment to implement these *biometric* authentication systems is very high. Due to the inherently intrusive and personal nature of the methods used, biometric systems have encountered significant user resistance as well as cost problems.

This paper describes an authentication scheme which has both a very low implementation cost and can be made totally unobtrusive to the extent that the user is not even aware that he or she is being authenticated. The method used is one based on a mechanical analogy of *graphology*, or handwriting analysis, by continuously analyzing in detail the *cadence* of the user's keyboard typing characteristics. The method is based on the premise that these characteristics will be unique to each user.

We classify this method as *semi-biometric*. The concept was originally proposed by Spillane in 1975 [3] and is based on extensive psychological studies [4,5].

It is worth noting that the system is implemented entirely in software, leading to low cost and easy portability across platforms.

IMPLEMENTATION STRATEGIES

If a continuous authentication system is to be used to determine the user's access rights to a system, it is clearly desirable that the system should determine whether or not the user is acceptable within as short a period of time as possible. In this way, if the user is an impostor, the amount of potential damage that might be

'European Convention on Security and Detection', 16–18 May 1995, Conference publication No. 408, © IEE, 1995

done to the system can be limited. A previous experiment [6] indicates that a well designed system can detect an impostor in less than 100 keystrokes. While this is impressive, it is always possible that the use may have temporary difficulties that alter his or her typing style slightly. For example, the room might be cold and the user keeps his heavy coat on which restricts his arm movement slightly!

The strategy chosen for this implementation is one of continuous authentication that does not forcibly log the user out after a pass/fail decision has been made. Rather, the system continuously logs the users typing characteristics and evaluates the likelihood of the user being who he claims to be. Should a serious discrepancy arise, the system could be used to warn the system administrator of a potential problem who can then the appropriate action.

KEYSTROKE CHARACTERISTICS

There are a number of different aspects of keystroke characteristics that can be used as identification criteria

- **Intervals between keystrokes.** These can be analyzed on the basis of a simple mean time interval across *all* keystrokes or between particular *pairs* of keystrokes of significance such as common pairs of characters (digraphs).

- **Duration of keystrokes.**

- **Frequency of errors.** This could be monitored by detecting the specific use of the delete and backspace keys.

- **Force of keystrokes.** While this might give valuable additional information, no computer keyboard offers the ability to measure this quantity.

- **Rate of typing.** The average number of words or characters per minute.

- **Statistics of text.** The individual language use or *style* of a user might be analyzed but this would require significant natural language processing and would only be applicable in those situations where a reasonably large amount of text processing was being carried out.

MONITORING SYSTEM

The PC keyboard has certain electrical characteristics that make the measurement of some keystroke characteristics easier than others. Of the characteristics mentioned above, the *time interval* between keystrokes and the *duration* of keystrokes are the easiest to measure as the keyboard generates hardware interrupts to the BIOS for each key *press* and each key *release*. In addition, the keyboard hardware automatically handles *n-key rollover* and *m-key lockout*. These two aspects are important for the keyboard in handling the natural typing characteristics of most users. Unless a user is particularly slow and uses "one finger at a time", it is natural when typing reasonably fast to sometimes press a second key slightly before releasing the first. It is the function of the *n*-key rollover circuitry to handle this situation and separate the pressing of one key from the release of a different key that would otherwise confuse the BIOS and lead to errors in data entry as well as user frustration with the computer. The *m*-key lockout circuitry supports this function by causing the depression of any keys after the first *m* keys to be ignored.

Within the hardware of the PC, the *programmable interrupt controller* assigns the various hardware interrupts (IRQ's) to corresponding software interrupts. We exploit this system by installing *interrupt handlers* on the software interrupt assigned to the *timer tick* interrupt (08h) and to the *keyboard* interrupt (09h).

TIMING ACCURACY

The timer interrupt normally occurs approximately 18.6 times each second and is used in various internal timing and housekeeping tasks such as updating the system clock and timing disk accesses. The timer chip, however, is programmable via the hardware ports and the timer tick can be set to any desired rate (less than the master CPU clock) by changing the divisor in the timer count registers. To facilitate accurate timing of the keyboard activity, the timer tick is increased to 10,000 ticks per second giving a timing resolution of 100 microseconds. It is vital, however, not to let these interrupts reach other parts of the system that rely on timing information for their correct operation. For example, the system clock would gain at an enormous rate and disk accesses would be thrown into chaos due to incorrect timeouts. The monitoring software therefore *chains* the interrupt through only once in every 10,000 interrupts. The rest of the interrupts simply trigger an IRET (interrupt return).

KEYBOARD MONITORING

The keyboard hardware interrupt occurs once for each key depression and for each key release. The hardware *scan codes* associated with each key are transmitted as well so that each physical key can be identified. It is important to note that much of this information is lost after processing by the BIOS. For example, capital letters can be obtained by pressing either *shift* key - there is no apparent difference between them. Likewise, most keyboards have two CRTL keys and two ALT keys to suit the convenience of both right and left handed users. The BIOS keyboard processor generates exactly the same output for both shift keys or CRTL

keys or ALT keys. The information as to whether the right or left shift key was pressed is discarded. While this makes no difference to the logical operation of most applications, it is useful to be able to distinguish between physically different keys in the current application.

The keyboard interrupt handler is called each time a key is pressed or released. The key associated with each operation is noted and the number of timer ticks between events is recorded. From this data, the duration of each keystroke, the interval between keystrokes and the overlap between keystrokes is computed. A running update of the mean and variance of these quantities is kept and is available for display in the demonstration system.

DEMONSTRATION SOFTWARE

The demonstration software is written in Turbo Pascal and is suitable for use on any PC from the earliest models through to the Pentium. The source code consists of a single file and uses the TurboPower library for some of the interrupt functions. The executable installs as a TSR and logs the keyboard activity.

The demonstration software is not designed as a comprehensive application but rather as a "bare bones" system to demonstrate the ideas behind the method. The source code is therefore very simple and easy to incorporate into more sophisticated systems.

The demonstration software uses the monitoring system in a small application that required the user to type the word "PASSWORD" five times. When the user has finished, the statistics of the session are displayed including the individual durations and intervals for each keystroke together with the overall mean and variance. In a simple real-world application, these data could be compared with a database of authorized users' typing characteristics.

Copies of both the source code and the executable are available for delegates to take away for further experimentation and development and incorporation into their own systems. (It is requested that delegates bring a floppy diskette if they require a copy of the files.)

RESULTS

The experimental systems gives good results. Typical key durations for an average typist are on the order of tens of milliseconds which allows their measurement to within 1% accuracy. Keystroke intervals can vary widely. Effectively "negative" times are possible where the second key is pressed before the first is released. The software uses the rollover capability to correctly determine the intervals in this case. At the other end of the timescale, very long periods can occur when the user pauses to think. During such pauses, the system can be temporarily suspended to prevent the running mean and variance from being badly skewed.

Just using the mean and variance as identification criteria, we have been able to distinguish between the typing styles of four users. Two of these are professional typists whose word-per-minute ratings are similar but who learned from different teachers. The third user is not a professional typist but a frequent keyboard user. The fourth is an occasional user. These results are summarized in Table 1.

Table 1. Summary of users typing characteristics

	---Duration---		---Interval---	
User	Mean	Var	Mean	Var
1	66ms	12ms	122ms	18ms
2	59ms	8ms	104ms	22ms
3	81ms	20ms	225ms	140ms
4	102ms	34ms	665ms	320ms

As can be seen from the table, the first two users are achieving around 6 keystrokes per second which corresponds to around 60 words per minute for an average word length of 6 letters. While their keystroke durations are comparable, their keystroke intervals are significantly different. The third user is somewhat slower with a much higher variance in interval as he "searches" for keys. The fourth user's lack of experience is quite obvious!

CONCLUSIONS

A demonstration continuous authentication software system using typing cadence has been developed for the IBM PC. The system is simple, low cost and gives accurate statistics of keystroke duration and interval. Initial studies show that these data are capable of distinguishing and identifying individuals. The software is a stand-alone module that can be incorporated easily into more sophisticated security systems. The source code is freely available to those who wish to experiment further.

REFERENCES

[1] Jobusch D.L. and Oldehoeft A.E., "A survey of password mechanisms : weakness and potential improvements", *Computers and Security*, **8**, No 7, 587-604.

[2] Wood H.M., "The use of passwords for controlled access to computer resources", *NBS*

Special Publication 500-9, US Department of Commerce, May 1977.

[3] Spillane R., "Keyboard apparatus for personal identification", *IBM Technical Disclosure Bulletin*, **17**, No 3346.

[4] Cooper W.E., *Cognitive aspects of skilled typewriting*, Springer Verlag, Berlin, 1983, 29-32.

[5] Shaffer L.H., "The basis of transcription of skill", *Journal of Experimental Psychology*, **84**, 1970, 424-440.

[6] Legget J, Williams G. and Usnick M., "Dynamic identity verification via keystroke characteristics", *International Journal of Man-Machine Studies*, **35**, No 6, 1991, 859-870.

BIBLIOGRAPHY

Dr Simon J Shepherd holds a First Class Honours degree in Electrical Engineering from the Royal Naval College Manadon and a Ph.D in cryptography from the University of Plymouth. He is a Member of the London Mathematical Society and a Fellow of the Institution of Mathematics and Its Applications. He holds a Commission in the Royal Navy where he served on the active list for twelve years and is now serving on the reserve list. He is currently Lecturer in Cryptography and Computer Security in the University of Bradford where his research interests cover all aspects of cryptography and computer and communications security. Dr Shepherd can be reached by email on `s.j.shepherd@bradford.ac.uk`.
