Keystroke Biometric Keyboard

December 7, 2010

Nathan Comstock (37593050), Sanduni Patikiriarachchi (82726043), Victor Tsang (79447041), Yee Chung Wong (37593050)

Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

mr.n.n.comstock@gmail.com, sandunip@gmail.com,victor.kc.tsang@ieee.org, joviw0715@hotmail.com

Abstract- **A preliminary design for a keystroke biometric keyboard is presented. Current commercially available keystroke biometric systems are based in software. This software is vulnerable to man-in-the-middle attacks, and sometimes requires a dedicated server to function. A hardware based keystroke biometric system overcomes some of the obstacles of software based systems to provide a more robust additional layer of security for users.**

Using an inexpensive microcontroller connected to a keyboard, a hardware prototype is constructed to illustrate the concept. A basic software suite is written, and a user profile is constructed. A method for comparing typing patterns is presented from an outside article, and implemented in the software. A comparison of false-rejection and false-acceptance rates is made, and the results indicate that more metrics must be added to improve usability. Finally, suggestions for future improvement upon the design are presented.

I. INTRODUCTION

Keystroke biometrics have been used, in one form or another, since the late 1800's. During that time, telegraph operators developed a particular rhythm or pattern when sending messages via Morse Code. The rhythm could be heard by the receiver, and these unique signatures were vital to determining if messages sent via telegraph were from friend or foe during WW1.

Modern incarnations of this concept allow for an extra layer of security to be put into personal computers, with similar effectiveness to iris or fingerprint scanning. When a user profile is created, that profile can be used to compare typing patterns for a variety of different metrics, giving the computer the ability to actively monitor its user and remove privileges when required.

Commercial vendors have come up with several interesting ways to implement this system in software, ranging from products that only authenticate at log in, authenticate continuously, require a dedicated server to function, etc. By implementing this system in hardware, it is shown how many of the difficulties and shortcomings of software systems can be overcome.

After construction, coding, testing and profile construction was complete, several user's typing "scores" were compared against the score of the person belonging to the profile. It is shown that the single metric implemented here would probably not be good enough for continuous authentication and more metrics should be added to improve both the false acceptance rate (FAR) and false rejection rate (FRR).

II. RELATED WORK

Currently, there is an existing software program, "Behavio" [1], which uses keystroke biometrics to actively authenticate users. In order for this program to operate, it must collect the user's keystroke information for at least one day of computer usage. After this process, the program creates a user profile which is used for comparing with the current user score. The program continuously computes a score based on the current user's typing patterns.

We have acquired a trial version of this program. After using it for a week, we have notice some flaws with this program. After installation, the program runs on the task bar without any notice on whether the program is running or not. Double clicking on the icon shows no response. We believe it is a security feature which is used to prevent an intruder from terminating the program. However, we can easily disable the program by going to the task manager and terminating the process. This program is not running until a user is logged on into the computer. As a result, keystroke biometrics are not used to authenticate a user when entering a user name and password.

"Behavio" depends on an online server to store the user profile. The computer must exchange profile information with the server before keystroke biometric can be used. Although this design allows users to move from one computer to another, there is a potential vulnerability if an intruder intercepts a user profile. We will try to address most of the shortcomings of software keystroke biometric with our hardware based keystroke biometric.

III. Solution

The two keystroke metrics used in this project are key dwell-time and flight-time. Dwell-time is the elapsed time between a key press and release. Flight-time is the time elapsed between the release of a key and the depression of the next key. Flight-time is a slightly less accurate metric, due to the fact that in continuous authentication the next key to be pressed is unknown. For this reason, flight-time is used in the preliminary score calculations for login authentication where the phrase is known, but not in the final score calculations for continuous authentication.

The prototype keyboard system is a PDA keyboard, with an internal logic controller, wired into an ARM microcontroller. The keyboard controller passes hex values for every key depression and key release to the microcontroller. The microcontroller translates these hex values in to ASCII values and passes them on to the computer via USB. In parallel, the microcontroller makes comparisons between the user's typing patterns and the profile that is currently loaded within it. This is done through the software stored inside the microcontroller. The microcontroller firmware is a derivative of the example USB firmware provided by ARM on their website [2] [3], modified to function like a USB keyboard.

The comparison of the keystroke data to the profile data happens after a key has been pressed and passed through. In this way, the microcontroller can lock the workstation when it has gathered enough data to be sure that the current user does not belong to the keystroke profile. This method allows for an improved and adjustable FRR, which in turn improves psychological acceptability.

The methods for gathering biometric data, creating a user profile and techniques in calculating scores to compare against the user profile were extracted from a paper written on keystroke dynamics [4], outlining a statistical fusion method for "scoring" a user against a profile. The fusion method includes a weighted average score based on Gaussian distribution and Direction Similarity Measure (DSM). For an in depth explanation, one should definitely read the paper, but a brief overview follows.

The user profile consists of a collection of mean and standard deviation (SD) values, the calculations for determining the mean and SD are shown in (1) below. Two individual profiles are constructed; one for continuous authentication mode and one for login authentication mode. For continuous authentication, the profile consists of the 11 most frequently occurring letters in the English language and their corresponding mean/SD. These frequently used characters are chosen in an attempt to simplify the profile comparisons and in turn improve the FAR/FRR. It was found that a relatively small data set of 500 words, typed by the user, is enough information to calculate an accurate mean and SD, more data does not change the values significantly. For login authentication, the profile consists of the mean/SD of the characters in the password, arranged sequentially. The password itself is stored as a hashed value. When a user types a phrase of equal length to the password, a hash of that phrase is made and compared against the password hash to determine if the phrases match and a score needs to be calculated.

TABLE I
Units for Equations

Symbol	Quantity
n	Number of training sample
μ	Mean of each individual character from n training samples,
σ	Standard deviation of each individual character from n training samples
D^i	Keystroke timing feature of each individual character (either dwell time or flight time)
Score	The Gaussian score
m	Total matches in a phase
c	Total characters in a phase
w	weight

$$\mu = \frac{1}{n}\sum_{i=1}^{n} D^i \qquad (1)$$

Source: Teh et al, Statistical Fusion Approach on Keystroke Dynamics

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(D^i - \mu)^2} \qquad (2)$$

The Gaussian score is a value from 0-1 indicating how well a specific timer matches the profile. One is a perfect match, with the dwell or flight time being same value as the profile mean. The equation for calculating the Gaussian score can be found in (3) below.

$$Score_{GD,D^i} = e^{\frac{(D^{i-\mu})^2}{2\sigma^2}} \qquad (3)$$

Source: Teh et al, Statistical Fusion Approach on Keystroke Dynamics

The DSM score is a measure of the typist's "rhythm", and compares key timing of specific words or phrases. This method is used for login authentication, where the phrase is known and stored as a hashed value [5] within the microcontroller. Figure 1 below shows how the DSM score is calculated. The two data-sets in the graph represent the user dwell times and the profile mean times. The variable M starts out equal to N-1 and every time the two graphs intersect M is decremented by 1.

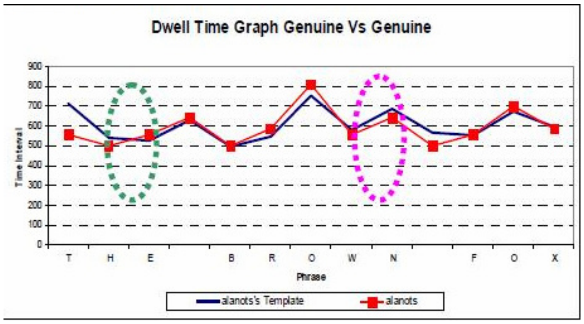

Fig. 1.

$$Score_{DS} = \frac{m}{c-1} \qquad (4)$$

Source: Teh et al, Statistical Fusion Approach on Keystroke Dynamics

The two scores, Gaussian and DSM, are weighted using the formula in (5) below, to create the final score. In practice *w* should be set to 1 for continuous authentication, to disable DSM scoring, and some value less than 1 for log in authentication.

$$Score_{Final} = Score_{GD} * w + Score_{DS} * (1-w)$$

Where 0<w<1

(5)

Source: Teh et al, Statistical Fusion Approach on Keystroke Dynamics

As explained above, the score in continuous authentication mode is calculated from the Gaussian score of dwell times only. A 3x11 array is populated; columns correspond to each of the 11 most frequently used letters in the English alphabet, the first row indicates how many times that specific character was pressed, the second row is a summation of all the Gaussian scores for that character and the third row is an average Gaussian score for that character calculated by dividing row two by row one. Predictably, the final Gaussian score is a summation of all the values in row 3 divided by the number of characters involved, 11.

The result of this profile creation, comparison and scoring is outlined in Figure 5 below. In continuous authentication mode, using only the Gaussian score of dwell time, five users type for two minutes. The typing profile belongs to Victor, who can be seen to have the highest score. The plot of Score vs. Time shows how the score becomes accurate and stable after approximately 20 seconds of continuous typing. The two users with discontinuities in their score plots, User1 and User2, were not familiar with the keyboard and the discontinuity can be attributed to this. The owner of the profile, Victor, can consistently get a score of 0.9 or higher while all the other users were unable to score higher than 0.88.

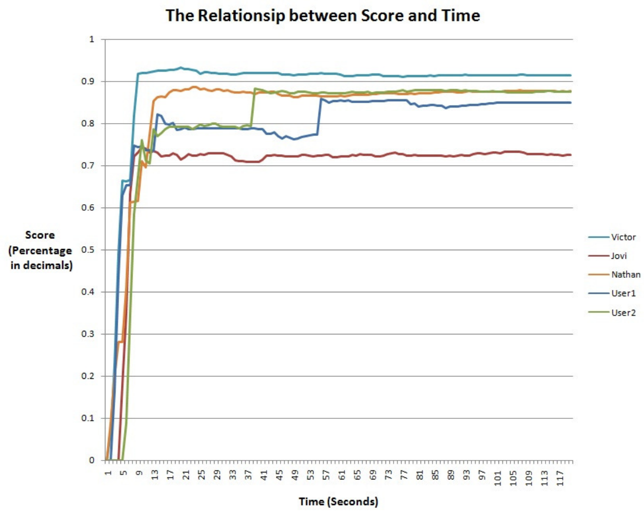

Fig. 2

IV. Discussion and Suggested Improvements

The results of the user comparison in Figure 2 show that the system does in fact work, and that Gaussian scoring of dwell times in continuous authentication mode is enough to differentiate the user belonging to the profile from other users. With a threshold score of 0.9, the FAR and FRR are both zero after approximately 10 seconds of typing. This data is not statistically significant due to the small sample size, but it adequately illustrates the concept.

The results of the login authentication mode testing with the weighted sum of Gaussian and DSM scores is not included in the report as the DSM scoring did not appear to function as outlined in the paper describing it. It is unclear if this is due to a misinterpretation of the paper or if the method is flawed. What the data does clearly indicate is that Gaussian scoring alone is not enough for accurate login authentication. The user scores are too heavily related to average typing speed for a small data set such as a user name or password. An intruder who knows a specific user's password could simply type the user name and password several times at different speeds with a very high likelihood of breaking the biometric security system.

With the short development time frame for this system, there is a lot of room left for future improvement. A list of suggested improvements, should this keyboard be developed as a commercial product, is as follows:

- More/improved keystroke comparison algorithms and metrics
- Improved functionality in the keyboard software/firmware
- Addition of a user interface, possibly via LCD
- Encryption of the data between the microcontroller and the computer

Of these suggestions, adding more metrics to the keystroke comparison software will have the most dramatic impact on the accuracy and usability of the system. It has been proven that users can be accurately differentiated by their typing patterns, but the limitation of having only one metric for comparison is likely to increase the FAR and FRR for a large sample size to the point where the system is simply not usable. Additional suggested metrics, beyond user scoring by dwell and flight times, are as follows:

- Use of left/right Shift key
- Improved "rhythm" algorithm
- Common typing errors
- Overall average typing speed
- Profile variation over extended periods
- Automatic detection and secure storage of commonly used user names/email addresses/passwords

The goal of a more advanced control algorithm for this system would be to establish the most balanced score values and improve the FAR and FRR. Ideally, the system would be developed to the point that the correct user would not notice the difference between a regular keyboard and this one, and an incorrect user would be logged out of the computer after ~30 or fewer keystrokes.

One way of improving the development time frame might be to implement a benchmarking system as suggested by Romain Giot *et al.* in their paper on the subject of Keystroke Biometric Benchmarking [6].

V. CONCLUSION

It has been shown that keystroke biometrics can be effectively used to differentiate between keyboard users, as all individuals have their own patterns and rhythms of typing. This project focused on capturing two main keystroke metrics, dwell-time and flight-time, to create a user profile and store it inside a microcontroller. The microcontroller compares the profile with the user typing behavior and calculates a similarity score to authenticate the user. Implementing the authentication inside the hardware itself eliminates the risk of MIM attacks while hashing of passwords inside the microcontroller ensures that, even with a compromised microcontroller unit, the impostor will not be able to easily extract password data. The two main design principles for secure systems used are psychological acceptability and defense in depth. The ease of use of this keyboard ensures that the security system adds no extra work for the user, increasing psychological acceptability. Actively monitoring the user to ensure correct access privileges adds an additional layer of security to the computer, improving defense in depth.

REFERENCES:

[1] BehavioSec (Reg. Bihaviometrics AB) (2010) Security through User Behaviour [Online]. Available: http://www.behaviosec.com/products

[2] STMicroelectronics. (2010, July 5). ARM-based 32-bit MCU STM32F10xxx USB Device Full Speed Library [Online]. Available: http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/um0424.zip

[3] STMicroelectronics. (2010, October 15). ARM-based 32-bit MCU STM32F10xxx standard peripheral library [Online]. Available: http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32f10x_stdperiph_lib.zip

[4] Pin Shen Teh *et al.* "Statistical Fusion Approach on Keystroke Dynamics," Multimedia University, Melaka, Malaysia and Yonsei University, Seoul, South Korea. Published in 2008

[5] Brainspark B.V. (2010, August 16). Polar SSL MD5 Library [Online]. Available: http://polarssl.org/download?file=code/releases/polarssl-0.14.0-gpl.tgz&name=polarssl-0.14.0-gpl.tgz

[6] Romain Giot *et al.* "GREYC Keystroke: a Benchmark for Keystroke Dynamics Biometric Systems," ENSICAEN, France. Published in 2009