ON USING MOUSE MOVEMENTS AS A BIOMETRIC

Shivani Hashia[a], Chris Pollett[b], Mark Stamp[c]
[a,b,c]Dept. of Computer of Science,
MacQuarrie Hall, San Jose State University,
One Washington Square, San Jose, CA 95195, USA.

ABSTRACT: *This paper considers the effectiveness of using mouse movements as a biometric. Two authentication schemes are proposed, one for initial login of users and another for passively monitoring a computer for suspicious usage patterns. Error rates for both schemes were calculated and compared to prior work.*

Keywords: *Authentication, Biometric, Mouse Movements.*

1. INTRODUCTION

In today's world, where many important tasks are done with a few clicks on the computer, the need for reliable, cheap, security systems is growing. The three most commonly used techniques for restricting access to a computer system are: passwords, smart cards, and biometrics. Each has its own advantages and disadvantages: Although passwords are cheap, if poorly chosen, they are not very secure. Further, as people are forced to remember more and more passwords, the tendency to write them down and other compromising activities increases. Smart cards and USB key devices are again reasonably cheap, and, though not as prone to attacks based on poor choices of key, are nevertheless easily misplaced, stolen, or lent in an ill-advised fashion. Further smart card reader deployment (as opposed to USB which is nearly ubiquitous) is also a non-trivial issue that must be dealt with effectively. In the biometric approach a physical or biological characteristic of a person is used to grant or deny a user access to a system. Often users find biometrics intrusive and many biometric devices are expensive. This report describes the use of mouse movements as a biometric. As mice are common to nearly all personal computers, a mouse biometric would be cheap and relatively unobtrusive. Unlike a smart card, this biometric is also easily deployed and not easily stolen, lost, forgotten, or lent.

A survey, commissioned by EDS and the International Association of Privacy Professionals (IAPP), and conducted by the Ponemon Institute found about 61% of the consumers did not want to be forced to change their passwords according to some fixed time schedule. About 66% accepted that it was more difficult to tolerate the inconvenience of failed authentication than being verified without proper authentication. Peter Reid, portfolio strategist for EDS Security and Privacy Services says "These findings are a clear indication that consumers are not willing to spend time with identification verification processes that serve to protect their personal information and their identity," [5]. So it is really important that the authorization process be easy to use and effective. The survey also showed that about 69% of the respondents were willing to use biometrics technology out of which 88% accepted the biometric technology because it was convenient to use and one did not have to remember passwords. [5]

Some commonly used biometrics include fingerprint, iris scan, palm print, facial recognition and gait recognition. A nice overview of some of the systems can be found in the report [3]. There are usually two stages for using a biometric in authentication: In the first stage, a person is enrolled. i.e., his unique features with respect to the biometric being used are extracted and a feature set is created for that user which is stored in a template file for him. The next phase is verification where the user later tries to log in to whatever system he wanted access to. Here again the user's parameters are measured with respect to the biometric. Then a unique feature set is determined and a comparison is done with the template file of the user. If they match, the user is verified to be the authentic user. In addition to these two phases, there may also be a training phase, where a number of samples are taken from the same user to make the verification more reliable.

Our mouse based authentication scheme makes use of this two-phased approach with additional sampling. The user enrolls in the systems by moving the mouse to follow a sequence of dots presented on the screen. Features are extracted from several repetitions of this task. During the verification phase, the user tries to login by moving the mouse on the same pattern of dots as were presented during the registration phase. The features we extract to verify users were determined by several experiments we performed and are described later in this article. These experiments allowed us to determine the error-rates of our authentication scheme and compare them with other biometrics and similar research that has appeared in the literature [1] [2]. The error rate we obtained is the preliminary result of our ongoing research on this subject. We hope to reduce the error rate with the new experiments that we plan to do in future.

The organization of the rest of this paper is now discussed. In the next section we provide a description of our mouse authentication scheme. We then give results of the various experiments we conducted to try to extract useful authentications features from this scheme. This is then compared with prior work and other biometric schemes. Additional experiments conducted on passive authentication are then briefly described. Finally, we give some concluding remarks.

2. DESCRIPTION

Our authentication model works in two phases: enrollment and verification.

2.1 Enrollment

When the user tries to logon onto our system, a login screen with start and stop buttons appears. When the user presses the start button, he is shown a first dot. The user is supposed to move the mouse to this dot. When the mouse reaches the dot, it disappears and the next dot at some other coordinate is drawn. The user has to follow the dots as they appear on the screen. He is shown ten dots with each dot appearing at different coordinates. The idea is to check how the user moves the mouse when he moves from one position on the screen to another. Based on the user's mouse movements, the coordinates of the mouse are recorded after every 50ms. Using these recorded coordinates, we calculate the speed, deviation from a straight line and angle. Deviation is the perpendicular distance from the point where the mouse is currently located (while moving) to the line formed between by the two points between which the mouse is moving. The angle we calculate is given by the angle formed between these three points. The angle is further separated into positive angle (an angle which lies between 0 to 180 degrees) and negative angle (an angle which lies in the range 0 to -180 degrees). These parameters are stored in a file for that user. The user has to move 20 times on the ten points to complete the registration. Between each pair of points, we find the average, standard deviation, minimum and maximum of the four parameters speed, deviation, positive angle and negative angle. Thus, there are sixteen parameters for a pair of points. So there are one hundred and forty four parameters stored in the template file for all nine successive pairs of points. After finding all of these parameters for a user, the parameters are then normalized, averaged and stored in a file (Vector.txt). We further calculate the standard deviation of each parameter from its average and store this as well as the user's name in another file (AvSd.txt). This file along with Vector.txt is used during verification phase when the user actually tries to login.

2.2 Verification

In the verification phase, we check if a user is who he claims to be based on Vector.txt and AvSd.txt files. To logon the user follows a dot pattern as in the enrollment phase. Each component of this login vector is checked to see if it lies within 1.5 standard deviations of the average from enrollment phase. If a parameter lies within 1.5* standard deviations, a counter is incremented by 1. This procedure is repeated for all 144 parameters. Then we check the value of this counter. We compare this value with the range of the user's counter value calculated during the registration. If the counter value was within the user's range, the user is authenticated else he is rejected.

3. RESULTS

We tested the model with 15 users. The testing was done in a classroom with students in the age group of 22-30. We used the same computer with the same mouse and mouse pad for all the students. This was done to ensure that all the parameters which could affect the accuracy of the system remain constant. We computed the error rate which is the point where false acceptance rate = false rejection rate. The error rate for this system was 20%. Instead of using the range of the counter value, we also did experiments using average and standard deviation of the counter. We checked if the counter value of the user was within 1 standard deviation of the average. This improved the error rate from 20 to 15%.

4. COMPARISONS TO PRIOR WORK

Two main biometric schemes related to ours have appeared in the literature: Nisenson, et al. [3] and Gamboa and Fred [1] [2]. In this section, we briefly describe their work and compare it to ours.

Nisenson, et al. [3] propose a biometric based on user keystroke patterns and a variation on the Liv-Zempel compression used here in the context of next keystroke prediction. The user enters a sequence of sentences and answers to open ended questions. The system then trains on this data and tries to identify a left out sentence from the training set. The system obtains an accuracy of around 95% +/-3%.

Gamboa and Fred [1] [2] propose a mouse based authentication scheme similar to ours. They build their system based on a web-based display and monitoring system that they have also developed. The user authenticates by completing a matching memory game while his mouse movements are being monitored. Cubic spline curves are fit to the sampled mouse movement points as the data is cleaned. Various types of velocity, jitter, pauses in motion, and angle of mouse motion are calculated, as well as, means and standard deviations to get a 63 dimensional feature vector. A greedy algorithm based on sequential forward selection is used to successively add to a list of features identifying a user, the best feature not yet considered. This algorithm proceeds until the equal error rate (ERR –the point at which FAR=FRR) no longer decreases. The training phase and testing phase based on a sample of 180 strokes taken from each user, half being used for each phase. Feature selection was done on sequences of 10 strokes from this sample. Each stroke takes about a second and estimates of the equal error rate in the case of 30, 60, 90 second user interactions were estimated. These were respectively about 1/50, 1/100, and 1/200.

In our mouse scheme, the typical user requires about 20 seconds of interaction to complete login verification. We can posit from the 30, 60, and 90 second data above that in such a setting Gamboa and Fred's scheme would achieve an equal error rate of 3 to 4%. This is considerably lower than the 15% rate that we have obtained. The likely reason for these differences is both that they are doing more sophisticated (and hence, computationally expensive) calculations with the mouse motions they record, and they are using some aspects of their memory game as well in figuring out if the user is genuinely the user – this latter part then is not measuring solely the effectiveness of mouse authentication. So it might be the case, that a more simplistic approach such as ours could be more useful with memory limited stand-alone devices.

5. PASSIVE SCHEMES AND OTHER EXPERIMENTS

Besides our mouse based login scheme, we have also done some work on passively monitoring mouse movements after the user is logged in. We call this passive authentication. The idea is at any point in time after the user is logged-in, if the user's mouse motion around the screen does not match his recorded background movements' features sufficiently closely, he will be forced to logout. The technique to perform this authentication is the same as with active authentication with the difference that there are no fixed dots on the screen. Instead regions around the screen are treated like dots and a pattern for the mouse movements around these regions during registration phase is recorded. A similar method is followed for the verification phase where registered data points are compared with the current mouse movements. During enrollment phase for passive authentication, we run the program that runs in the background to record the mouse coordinates for 15 minutes. We separate the coordinates into dense regions i.e., we draw a convex polygon around the regions where we find there are more than 3 coordinates recorded within a 10-pixel range. That gives us all the regions where the mouse moves most of the time. We call them states. We then find the transitions from one state to another. We calculate the speed while moving from one state to another and also the wavering in the mouse when the user is in the same state. By wavering we mean the distance from each point in the transition state to the best fit line formed by all the points in the state. We then calculate the average speed, standard deviation of speed, average wavering and standard deviation of wavering. We store the transition state, count of how many times the user was in that state for those 15 minutes, average speed and average wavering in a file which is used during verification phase. During verification, we continuously keep recording the mouse coordinates in the background. Every two minutes, we read the recorded coordinates, find the speed and wavering when the mouse moves within the same state. We ignore all the other coordinates. We compare the parameters of speed and wavering with the parameters we found during registration phase. We check if the speed and wavering for a transition state are within the range of $\text{ave}_v \pm 1.5^* \sigma$ of the speed and angular velocity found in that transition state during registration. If they lie between that range, we keep on doing the same process, but if at any point of time, we find that the speed and wavering of the mouse within the last ten states do not fall within the specified range, we conclude that he is unlikely to be the actual user. From the preliminary experiments we could infer that it was after approximately 5 minutes that an actual user was considered an intruder and 2 minutes for which an intruder was allowed to work on the system.

6. CONCLUSION

Security plays a very important role in the modern world where almost everything is done with the computer. To make personal computers secure various biometrics have been developed. In our research, we attempted to develop one such model. We developed a model that can authenticate a user with his mouse movements. The model does not require any additional hardware.

In this authentication technique, there is an initial login where the user is presented with a screen and he has to move the mouse towards the dots drawn on the screen. The parameters that we used for this authentication are speed, deviation from straight line between two points and the angle of deviation. We made 144 parameters from these parameters and then used them to find uniqueness in user's movements.

Our eventual goal is to have a system that works on a broad range of devices from desktop computers to wall mounted touchpads. At the latter extreme the computational resources are limited. Even with limited resources our model can be further improved. We could always add some more parameters. The increase in the number of parameters will help to reduce false acceptance rate as well as false rejection rate. One more parameter that we could add is mouse clicks or finger taps. When the user reaches a dot on the login screen, we could require him to click when he thinks he reached the target dot instead of just making the dot disappear and the next dot appear. We could also note the response time of the clicks. Another idea in the desktop setting is to allow the user to select from different themed connect the dot patterns – shaped like favorite cartoon characters, or other familiar objects. Although this wouldn't necessarily improve the authentication, it could improve the user experience. Finally, we are in the process of continuing to develop our passive authentication scheme.

7. BIBLIOGRAPHY

[1] Gamboa, H., and Fred. A., "An Identity Authentication System Based on Human Computer Interaction Behaviour", *3rd Workshop on Pattern Recognition in Information Systems PRIS 2003.* pp. 46 –55.

[2] Gamboa, H., and Fred, A., "A behavioral biometric system based on human computer interaction", *Proceedings of SPIE -- Volume 5404.* Jain A. K., Ratha, N.K.,Editors. pp. 381–392. 2004.

[3] Nisenson, M., Yariv, I., El-Yaniv, R., and Meir, M., "Towards Behaviometric Security Systems: Learning to Identify a Typist", In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pp. 363–374, 2003.

[4] Wayman, E.J., National Biometric Test Center Collected Works, San Jose State University, Aug 2000. **http://www.engr.sjsu.edu/biometrics/nbtccw.pdf**

[5] FindBiometrics.com. Survey Finds Consumers Open to the Use of Innovative Identification Methods Such as Biometrics. **http://www.findbiometrics.com/Pages/feature%20articles/innovative-uses.html**

Biography

Shivani Hashia received her B.E. degree in Computer Engineering from College of Engineering, Pune India, in 2001 and the M.S degree in computer science form San Jose State University California in 2004. She is currently working at Amdocs. Her interests include network security and data mining.

Chris Pollett grew up in Canada. He has lived in California since his undergraduate days at Caltech. He obtained his Ph.D. in Mathematics from UC San Diego in 1997. Dr. Pollett currently has over twenty publications, mainly in the field of computational complexity and its interactions with mathematical logic.

Mark Stamp has been doing security for more than a dozen years, including 7 years at the NSA and 2 years at a small Silicon Valley startup company. For the past 3 years Dr. Stamp has been teaching security classes at San Jose State University. He recently completed a textbook, Information Security: Principles and Practice (Wiley, 2005).