

Mouse Movements Biometric Identification: A Feasibility Study

Adam Weiss, Anil Ramapanicker, Pranav Shah, Shinese Noble and Larry Immohr,
Seidenberg School of CSIS, Pace University
1 Martine Ave, White Plains, NY, 10606, USA
[aw72754w](mailto:aw72754w@pace.edu), [ar95556w](mailto:ar95556w@pace.edu), [ps67064p](mailto:ps67064p@pace.edu), [sn91927w](mailto:sn91927w@pace.edu) }@pace.edu
LarryImmohr@cs.com

Abstract

In this study we test the feasibility of mouse movement biometrics. Building on ideas from previous work in mouse and keystroke biometrics, the study focuses on defining data collection, feature metrics, and classification as a precursor to additional research. A small training set of twenty-five data samples from five users was used to develop and test the software. Data capture, features, feature extraction, user profile creation, classification, and the experiment are all discussed. The experiment was successful using Next Nearest Neighbor. A section on recommendations for future improvements and research is included.

1 Introduction

Why do a feasibility study on mouse movement biometrics? The security landscape is changing with internal threats and financial motivations replacing the activities of “script kiddies” seeking bragging rights about the number of machines compromised [5, 13]. Unauthorized access, Theft of proprietary information, and Insider Net abuse are within the top five financial losses [5]. There was an increase in the percent of respondents reporting that they did not know if unauthorized access had occurred [5]. Telecommuting and extranets are extending where the corporation ends as well as introducing multiple access points. Security experts see a need to go beyond the password and have multiple forms of identification [8]. Additionally, new approaches to handling security are needed [12]. Therefore, research is needed.

Biometrics research has been around for a long time, but has not seen wide implementation in the security field. The FBI report shows use of biometrics at 15 percent versus 97 percent adoption of firewall technology [5]. This low implementation rate may be the result of costs, lack of trust in the technology, or people’s reluctance to use the required devices. Fortunately, new research is leading the way to the use of biometrics [8]. Work with multi-modal biometrics has shown the value of authentication using personal characteristics [9]. There are two types of biometrics physiological and behavioral [4]. Fingerprints are considered physiological and keystrokes are behavioral. Both have the added advantage that users cannot easily leave them behind. Keystroke biometrics use existing equipment, the keyboard, and appear to be minimally invasive. Keystroke biometrics work on the basis of multiple feature extraction being used to create a profile of an individual [3, 6, 11, 13, 14]. This profile is used to identify or authenticate the user.

Another area of biometric research is mouse movement. Mouse movement has the two advantages of low cost and low invasiveness. Research has already been performed in the area of mouse movement biometrics [1, 2, 7, 10]. This initial research has shown promising results with the use of a specialized pressure sensitive mouse [7] and the use of mouse movements [2, 10].

A team from Perdue University and Tufts University recently took on a project where they wanted to see if it was possible to re-authenticate users through their mouse movements. This project is very similar to our research in that they are pulling in data for profiles through the use of individuals’ mouse movements and

their consistencies. Their underlying hypothesis was that they could successfully model users' behaviors on the basis of user-invoked mouse movement [2]. Normally the authentication is processed in the beginning of a session, however once that session is started; there is no way to find out if the user is who they say they are. One method of re-authentication is by monitoring the mouse movements of the user and comparing it to a profile [2]. Our study will be utilizing the same information this group used. We will also be looking into obtaining profiles of users and making comparisons through the nearest neighbor method.

In this feasibility study, a software system is created to gather mouse movement data; compute measurements for a set of features and use them to identify an unknown user from a known set of users. Metrics such as mouse moving speed, number of clicks, duration of clicks and variation in mouse trajectory (or arc) are under consideration to create a profile (signature) of each user.

The paper is structured as follows; Section 2 describes the mouse movement biometric system. Section 3 describes the experiments that we performed. Section 4 presents the results of those experiments and Section 5 presents our conclusions, discusses problems we encountered, and future recommendations.

2 Mouse Movement Biometric system

The mouse movement biometric system consists of three components: mouse data collection, feature extraction and pattern classification.

2.1 Data Capture

A Java standalone application is developed to collect mouse movement data from the user. When user start using the mouse for accomplishing his/her tasks, a monitoring program running in the background gathers the data. There are many user interfaces to this program. A data collection-training program with 25 buttons arranged in a 5X5 matrix is shown in figure 1.



Figure 1: Java application for data collection

In this data collection method, the user is required to click the buttons as guided by the program. Each time the user clicks and moves the mouse a background program collects the mouse movement and click data. The raw data file collected by the application is stored in a .csv file and contains the following information for each entry:

- Mouse event, whether it is a move, drag or click
- Time of the event in milliseconds
- X and Y co-ordinates of the mouse pointer on the user screen

Upon the completion of the task, a text file is generated which is delimited by ‘,’ character. A short version of the sample data file is shown in figure 2.

User screen Size	width	1680	height
timer	10		
mouseMoved	1162463438140	120	39
mouseMoved	1162463438156	49	3
mouseMoved	1162463438187	49	5
mouseMoved	1162463438203	49	6
mousePressed :left click 1 time(s)	1162463438375	49	6
mouseReleased	1162463438546	49	6
mouseMoved	1162463438640	58	8
mouseMoved	1162463438656	62	9
mouseMoved	1162463438671	140	51

Figure 2: A sample data file

2.2 Feature Extraction

We believe that, like keystroke biometrics; there is enough individuality in mouse usage to identify the user. In our initial investigations, we identified several features, which can be used to create a pattern, and then these patterns can be used to create a profile. We considered mouse movement and left mouse click as two mouse events for our preliminary experiments.

Feature Extraction involves taking the raw data that we collected in the Data Capture phase of the program and applying calculations to extract characteristics that signifies user behavior. From those measurements we create a feature vector, which in turn represents a user profile or a user signature. The following section describes feature definitions and how they are computed.

2.2.1 Mouse Movement System Features

The raw data collected from the data capture module is processed to create mouse curves and mouse clicks. Each curve and click is associated with a set of features namely size of the curve, length of the curve, speed of the curve, acceleration of the curve, duration of the click and curvature of the curve.

2.2.1.1 Size of a curve:

Size of a curve is defined, as the total number of continuous points constitutes the curve.

$$\text{Size of the curve } n = \sum_{i=4}^n (p_i) \quad (1)$$

p is a mouse data point. The curves less than four points are ignored.

2.2.1.2 Length of a mouse curve:

Length of the mouse curve is defined as the sum of the distances between all adjacent curve co-ordinates. A mouse curve c with n points has a length of:

$$\text{Len}(c) = \sum_{i=2}^n \sqrt{((x_i - x_{i-1}))^2 + (y_i - y_{i-1})^2} \quad (2)$$

2.2.1.3 Total time of the mouse curve:

Total time of the mouse curve is defined as the total time taken to complete the mouse curve. A mouse curve c with n point has a total time of:

$$\text{Total Time}(c) = \sum_{i=2}^n (t_i - t_{i-1}) \quad (3)$$

Where t is the time stamp on the mouse point.

2.2.1.4 Mouse speed over a pre-defined action

Mouse movement is obtained when the user moves mouse from point A to point B without pressing any mouse button. One movement ends when there is no input from the mouse for a pre-defined period. Drag and Drop is defined when the user moves mouse from point A to point B while pressing the left mouse button. The data capture module is having the facility to collect the data at different time intervals. We compute the mouse speed from the distance traveled and the time takes to reach the next point. The mouse speed may not be uniform through out the move. At the starting of move, it may at a low speed and at middle of move it is accelerated to a faster speed. With these characteristics each user creates a unique profile on his/her mouse movements.

Average speed of the curve: Average speed of the curve is defined as the average speed taken to complete the curve. The speed between two points is computed as the distance traveled over the time. A mouse curve c with n points has an average speed:

$$\text{Average speed}(c) = \frac{1}{n} \left(\sum_{i=2}^n \left(\frac{\sqrt{((x_i - x_{i-1}))^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \right) \right) \quad (4)$$

2.2.1.5 Angle of mouse movement or movement in a direction

Another interesting feature to consider is the angle of movement along with the speed of the movement. Depending upon the movement of mouse in a particular direction, the speed of the mouse can vary.

2.2.1.6 Acceleration

Acceleration is the time rate of change of velocity with respect to magnitude or direction, which is the derivative of velocity with respect to time [6]. Reading from the mouse movement's literatures [1] [2] indicates that as we move from one button to the next there will be an acceleration and deceleration of the mouse. We have taken acceleration as the time and divide it by velocity.

2.2.1.7 Mouse click duration

Mouse click is obtained when the user presses the left mouse button and releases it.

The duration of click is the time difference between the mouse press and mouse release events in a click. Due to the effect motor skills of a person, there can be significant difference on click duration of different personal.

Once each mouse click and curve measurements are computed, a user mouse profile is created using the mean, average and standard deviation of the all the individual features. This computation results in a feature vector. The following measurements are included in a feature vector:

- a) The average and standard deviation of these click durations
- b) The average and standard deviation of these transition times
- c) The average and standard deviation of the curvature measurements
- d) The average and standard deviation of the transition velocities
- e) The average and standard deviation of the transition accelerations

2.3 Classification

The purpose of classifier to check the validity of the features extracted and recognizing the patterns involved in the mouse movement characteristics of a user. We applied k-nearest neighbor method to identify unknown mouse profile from a set of known user profiles. The nearest neighbor algorithm is a simple classification algorithm. The test data set is classified according to the classification of nearest neighbor from a database of known classification, ie the training set. In the general version of this algorithm namely k-nearest neighbor, it outputs k nearest samples from the training set. In our implementation we used Euclidean distance to find the nearest neighbor. The feature extraction module provides an n-dimensional feature vector of a user. This n-dimensional feature vector is used to compute the distance an unknown entity and a set of known entities. In N dimensions, the Euclidean distance d between two points p and q is:

$$\text{Distance } d = \sqrt{\sum_{i=2}^n (p_i - q_i)^2} \quad (5)$$

Where p_i (or q_i) is the coordinate of p (or q) in dimension i.

The classifier program takes feature vector of a user as input. The feature vector is then normalized to create a normalized feature vector with following formula. The normalized feature x' of a feature x is:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (6)$$

where min and max are the minimum and maximum of the measurement over all samples from all subjects. This provides measurement values in the range 0-1 to give each measurement roughly equal weight.

The classifier program can classify the normalized feature in two different methods. In the identification method, the unknown test can is classified against a set of known user profiles. In leave-out one method a cross validation is done for all the files in the training set. In leave out one method one file compared against rest of the files and the process is repeated for all files.

3 Experiments

We gathered 25 data sets from 5 different users. Each user is allowed to use the training program 5 different times and allowed to do the same task. The training program comprised of 25 buttons screen arranged in a 5X5 matrix fashion. The buttons are enabled for click in a particular order. The administrator

configures this order and all the users used the same order. The order allowed us to obtain mouse curves of different lengths; some curves are long and some curves are short.

The data set obtained from the training program is the input to the feature extractor module. The feature extractor module parsed the raw data in to mouse curves and mouse clicks. The number data points in each curve are different depending upon the length of the curve. Each raw data file provides many curves and clicks to one user and each user is having 5 such files. The feature extractor module then computed the features described in section 2 for each data file. Measurement file is created from each of the 25 raw data files. Part of a measurement file is shown in figure 3.

type	points	length(pixels)	total time(msec)	average velocity(pixels/msec)
mouse curve	33	518.7206527	578	1.047530892
mouse curve	37	897.4416396	593	1.592189283
mouse curve	34	1035.443331	781	1.980847772
mouse curve	24	659.1251356	406	1.828141033
mouse curve	35	974.2207411	610	1.792954244
type	click Duration(mSec)	average Curve Time	average Speed	average Click Duration
mouse click	188	595	1.613133476	150
mouse click	172	SD Curve Time	SD Speed	SD Click Duration
mouse click	172	176.5227773	0.380001517	26.46087258

Figure 3: Part of a measurement file showing individual curve and click features.

The extracted features are then processed for the averages and standard deviations to create a feature vector. Part of the feature vector file as shown in figure 4

User Name	Average Curve Speed	SD Curve Speed	Average Curve Time	SD Curve Time	Average Click Duration	SD Click Duration
Anil	4.802340007	2.380358866	493.3076923	194.2780442	171.88	20.21138293
Anil	4.495298795	1.890681976	535.5384615	198.8965599	142.4	20.78637374
Anil	5.080432469	3.606432007	533.6923077	215.6319978	151.68	20.84866103
Adam	6.136556449	3.332073671	506.8148148	457.6513901	90.69230769	17.79788233
Adam	5.095276271	2.273071277	457.5	279.8532912	91.8	19.33709733
Adam	5.923297085	3.9417567	425.71875	286.3989998	92.19354839	12.64434933
Larry	3.704071254	1.643489258	713.0384615	364.8612128	109.72	23.3425934
Larry	4.297164069	1.761533948	595.4615385	489.594955	106.56	24.57641145
Larry	4.297164069	1.761533948	595.4615385	489.594955	106.56	24.57641145
shinese	4.60672646	2.626070174	637.9615385	448.4628653	89.68	47.55463525
shinese	4.868570464	4.291088222	605.8076923	562.4699322	68.84	46.7629483
shinese	4.082350251	2.254433552	933.4074074	1507.808505	96.30769231	41.96753324

Figure 4: Part of the feature vector file showing average and standard deviations of features.

The feature vector file is taken as an input to the classification process. We have done two classification experiments a) identifying an unknown test case from a user against the training set and b) a cross validation using leave out one method.

In the leave out one method one training file is validated against the other entire training set data files. The k-nearest neighbor algorithm is configured to input first 10 nearest neighbors. A sample classification process result for one case is shown in figure 5, this is repeated for all other cases.

Testing:Adam			
matching with the closest user:Adam placed at 1			
Adam	0.216764		
matching with the closest user:Adam placed at 2			
Adam	0.357693		
Larry	0.419064		
Larry	0.441511		
Anil	0.571657		
matching with the closest user:Adam placed at 6			
Adam	0.581672		
Larry	0.616915		
Larry	0.677445		
Larry	0.684775		
matching with the closest user:Adam placed at 10			

Figure 5: classification process result shown for one case

4 Results:

The classifier result from the leave out one method is further analyzed for success rate. The result file is shown in figure 6. In this study there are 25 data files, 5 files each from five users. In our experiment we obtained a success rate of 92 % for the first choice of the nearest neighbor. Matching the second choice was 88% and matching second and third choices together 80%.

CaseDescription	totalCases	matching	not matching	percentage
Matching first choice	25	23	2	92
Matching second choice	25	22	3	88
Matching first and second choice together	25	20	5	80
Matching third choice	25	14	11	56

Figure 6: Results for 25 files training set

5 Conclusions and recommendations

In this mouse movement biometric feasibility study, we received 92% success rate for the first nearest neighbor in the leave-one out method with a small size feature vector. The stand-alone system we built is very easy to operate and have many features that can be used for further studies in this area.

We recommend solidifying the results by conducting many more experiments. The experiments we have conducted are based on a fixed pattern for all users, we suggest conducting experiments with random patterns so that the data set size is not uniform. We also suggest creating a user profile from multiple files from the same user and testing for identification. This type of arrangement will provide much larger data sets (may be in the order of 100 curves). The feature extraction module has the facility to accept multiple files to create a single profile. Using different user interfaces may also be a good idea. Finally since the mouse movement biometric application is intend to use for online, we recommend to upgrade the standalone system to an on-line system.

6 References:

[1] Ahmed Awad E. Ahmed, and Issa Traore, Detecting Computer Intrusions Using Behavioral Biometrics, Department of Electrical and Computer Engineering, University of Victoria, P.O. Box 3055 STN CSC Victoria, B.C. V8W 3P6 CANADA

[2] Carla E. Brodley and Maja Pusara, User Re-Authentication via Mouse Movements, Department of Computer Science, Tufts University, Medford, MA 02155 and School of Electrical and Computer Engineering Purdue University, West Lafayette, IN 47906-2035

[3] D. D'Souza, Typing Dynamics Biometric Authentication. October 2002.

<http://innovexpo.itee.uq.edu.au/2002/projects/s373901/thesis.PDF>

[4] Ross A.J. Everitt and Peter W. McOwan, Java-Based Internet Biometric Authentication System, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 25, NO. 9, SEPTEMBER 2003

[5] Lawrence A. Gordon, Martin P. Loeb, William Lucyshyn and Robert Richardson, 2005 CSI/FBI COMPUTER CRIME AND SECURITY SURVEY

[6] Daniele Gunetti and Claudia Picardi, Keystroke Analysis of Free Text, University of Torino

[7] Curtis S. Ikehara and Martha E. Crosby, User Identification Based on the Analysis of the Forces Applied by a User to a Computer Mouse, University of Hawaii at Manoa, Department of Information and Computer Sciences

[8] Dave Kearns, Is biometrics the best solution to replace password authentication,
<http://www.networkworld.com/newsletters/dir/2005/0124id2.html>

[9] Sandeep Kumar Terence Sim Rajkumar Janakiraman Sheng Zhang, Using *Continuous* Biometric Verification to Protect Interactive Login Sessions, School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543

[10] Douglas A. Schulz, MOUSE CURVE BIOMETRICS, Pacific Northwest National Laboratory, U.S. Department of Energy

[11] M. Villani, M. Curtin, Ngo J. Simone, H. St. Fort, C. Tappert. S. -H Cha. Keystroke Biometric Recognition Studies on Long Text Input over the Internet, CSIS Pace University, Pleasantville, New York 10570

[12] Symantec Internet Security Threat Report Trends for January 06–June 06

[13] Verification via Keystroke Characteristics of Typed Names using Neural Networks,” International Journal of Man-Machine Studies, 39(6): 999-1014, 1993

[14] R. Gaines, W. Lisowski, S. Press, N. Shapiro, Authentication by keystroke timing: some preliminary results. Rand Report R-256-NSF. Rand Corporation; 1980