

# Mouse Biometric Authentication

Francisco Betances, Adam Pine, Gerald Thompson, Hedieh Zandikarimi, and  
Vinnie Monaco

*Seidenberg School of CSIS, Pace University, White Plains, New York*

**Abstract** — Increased security concerns within the computing world have forced security-minded users and developers to push for greater biometric verification techniques. The use of a mouse as a biometric verification device through the identification of unique user movements has gained traction and support within the industry. Although there is interest in the subject, there is question to the viability of such a concept and its ability to withstand unauthorized access attempts by malicious users. Studies within the industry have shown that while viability exists, implementation for end-user customers may be more difficult than initially planned.

**Index Terms** — Authentication, Mouse Biometrics, Mouse Clicks, Mouse Velocity

## 1) INTRODUCTION

The development of advanced computing systems and the transition of major industrial, commercial, and consumer level tasks to a purely computing environment has created a great concern in terms of computer security. While passwords and traditional methods of computer security are effective, they are unable to verify the biometric identity of the user; this creates a weakness that can be exploited through various access attempts to unverifiable access tokens.

The idea of using a mouse or touchpad device as a means of user authentication has potential as a method of biometric security, as such inputs would be difficult to imitate by other individuals. This is based on the premise that individual mouse movements are unique to the individual, as no two users engage with computer interfaces in the same manner. To support this research, a number of studies have shown promising development through the identification of unique user patterns toward computer behavior. As such, mouse dynamics serve as a potential behavioral dynamic that would be difficult to imitate without direct knowledge of user activities.

Mouse dynamics has its origins in the concept of keyboard dynamics, in which the identification of user-specific typing rhythms is used as a means of verifying user identity. From this perspective, mouse dynamics are a natural next-step in the process and can be used in combination with other behavioral biometric methods to

verify user identity.

While keyboard dynamics has been studied for several decades, the use of mouse dynamics in this concept is a recent phenomenon and remains largely untested, as it has not faced a similar type of scrutiny from the security community. However, several studies have reaffirmed that there is a high success rate in identifying users with a very low rate of false positives and false negatives [1]. Although successful, there are various methodologies for the implementation of the biometric readings with degrees of accuracy. Most important, the need for biometric authentication through behavioral readings must be able to identify users while preventing imitators from gaining successful entry into the system.

Because of the limitations of mouse dynamics and authentication, it has been argued that this form of biometric identification is best equipped in a form that complements a primary method, rather than relying on this type of method for the primary identification method.

## 2) MOUSE AND BIOMETRICS BACKGROUND

### a. Mouse Device Background

A mouse is synonymous with computer use but not many people know exactly how it works. A mouse is a pointing device that functions by detecting two-dimensional motion relative to its supporting surface [2].

The mouse device is usually made up of two buttons and a scroll wheel, which can also work as a third button. The left button is used for selecting items for dragging and dropping, and it is also used for pressing buttons. By clicking the right button, a user can access editing properties that the file, webpage or application may have. For mouse devices that have a scroll wheel, this can be used for scrolling up or down on a webpage, an application or a folder. The scroll wheel button also can be pressed and in certain situations it will automatically scroll the page in the direction that mouse is moving, and this stops by clicking the scroll wheel again. In certain games, it can be mapped to do another function and can be used like a regular mouse button or keyboard button for accessing items or certain objects in the game.

### b. Biometrics Background

Biometrics is a way to identify people based on a particular distinctiveness or a certain pattern. These patterns or distinctiveness can be a physical trait or another

noticeable characteristic that can be used to identify the user. There are certain types of characteristics that are used to identify a person, this includes their fingerprint, retinal scan, DNA, and facial or voice recognition. Based on the fact that biometrics is based on a particular identifiable trait, it is a good way to identify a person and as far as computer security goes, be able to grant access to that person.

There is usually some type of sensor that will get the information based on the given trait and check it against a database of what the characteristic should be. For physical traits, a camera can be used as a sensor because physical characteristics can be noticeable. For biometrics through computers, an application can be used that either logs keyboard strokes or mouse movements. Once the data is gathered, the acquired data is checked against the database to determine if the biometric data, in fact, matches the individual.

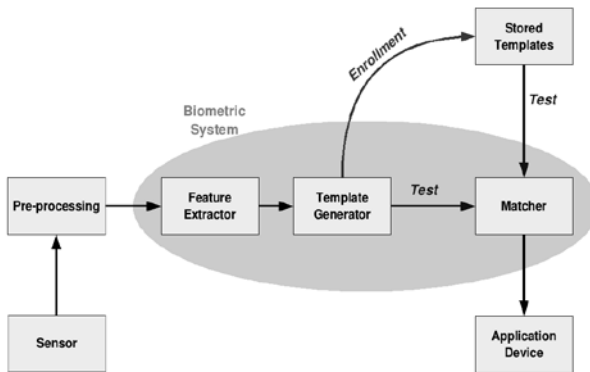


Fig. 1. This image shows the steps necessary for biometric authentication [3].

### c. False Acceptance and Rejection Rate

When doing a study on behavioral biometrics, and user verification using a mouse, it is imperative to study peers research on the topic. This has recently become a topic of great interest, and it is important to examine current research. Many good ideas have been developed on this topic, but there are still improvements to be made. When looking at two related research papers, it is not clear what the most effective method would be to test for a proper validation system.

Biometric systems are typically evaluated with the following variables [6]:

- *False Acceptance Rate (FAR)* - the probability that the system will incorrectly label the active user as the same user that produced the enrollment signature.
- *False Rejection Rate (FRR)* - the probability that the system will incorrectly label the active user as an impostor, when in fact it is not.
- *Equal Error Rate (EER)* - the error rate when the system's parameters (such as the decision threshold) are set such that the FRR and FAR are equal. A lower EER indicates a more accurate system.
- *Verification time*- the time required by the system to collect sufficient behavioral data to make an authentication decision. Because there can be significant pauses in the

data, the quantity of input data is often used rather than time.

## 2.1 Related Research

### a. Identifying Game Players – University of Washington

The University of Washington worked on identifying game players using mouse movement in two popular video games, Solitaire, and StarCraft [4]. The process of collecting data is to perform the baseline experiments, playing Solitaire, and playing StarCraft. The players were all female and the experiment ran on the same computer to ensure that all the parameters were consistent for all the users; in contrast to our data collection we gather data using different computers. In the baseline experiment, an application was developed to gather data in a controlled environment for each individual; this application could capture three major mouse actions: mouse moves, clicks, and drags. The first task requires users to click rapidly and accurately between two targets, the second task requires the users to drag a circular shape in a specific range, and the final task requires users to double click on a target. In the Solitaire and StarCraft experiment all the users play the game individually and then the data is collected.

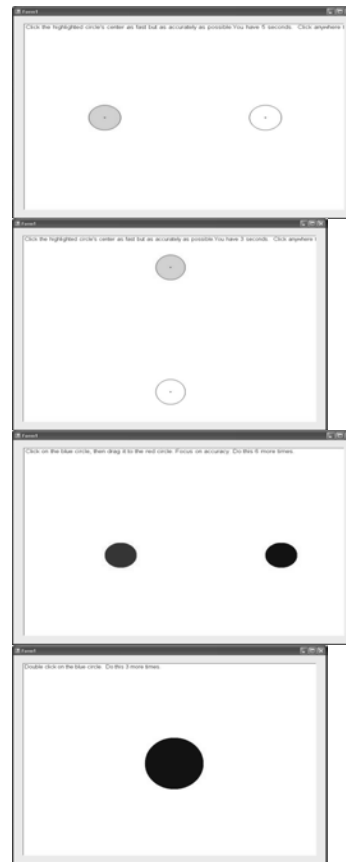


Fig. 2. Baseline program, clicking tasks, dragging tasks, and double clicking tasks [4].

In their experiment, they developed a C# program to log low-level mouse movements.

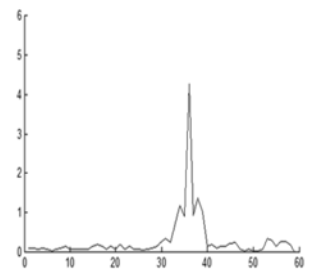
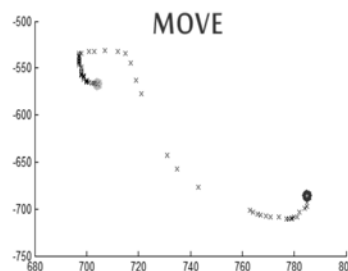
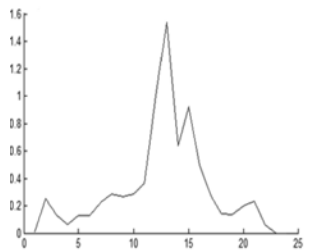
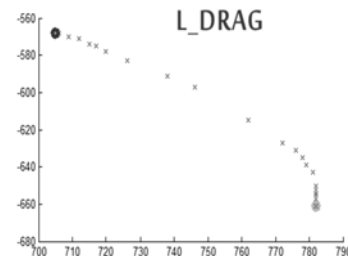
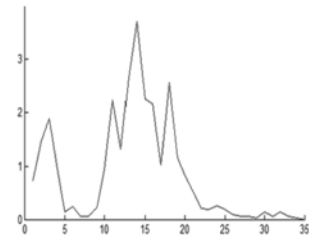
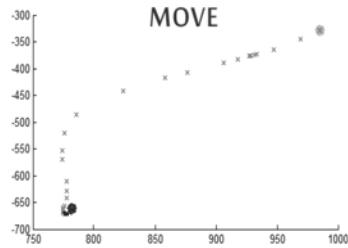


Fig. 3. Visualization of the first three movement in a Solitaire game. First column: location of a mouse event comprising the action. Second column: normalized velocity for each mouse event [4].

After gathering the data from the players, they used SVM, 1-Nearest Neighbor, and 7-Nearest Neighbor models. The Neighbor's model work more accurately than the SVM model according to their experiment. The models they constructed do not perform well across a game domain. However, it is accurate enough to identify cheating players or unauthorized users.

#### b. Authentication Methods – San Jose State University

San Jose State University worked on mouse movement as a biometric. They proposed two authentication methods, one for initial login of users and another for security purposes to monitor a computer for suspicious usage patterns, basically their authentication models works in two phases: enrollment and verification [5]. The user enrolls in the system by moving the mouse to follow a sequence of dots presented on the screen. 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 purpose of these experiments is to calculate the error rate of their authentication scheme and compare with other biometric research.

In the enrollment phase the user logs into the system, and they're supposed to move the mouse towards the dot that appears on the screen, click on it, and the dot will disappear. This process has to be repeated ten times. Based on the user's mouse movements, the coordinates of the mouse are recorded. Speed, deviation from the straight line and angles are calculated. The data collected in this phase is being used in the verification phase when the user tries to log in. In the verification phase, the system checks to see if the user's credentials are correct based on the data collected in the enrollment phase. To log in, the user follows the same pattern as the enrollment phase. In their scheme, it takes 20 seconds for the user to complete login verification. The result from this phase will be compared with the result calculated during the registration.

This model had been tested on 15 users all using the same computer to ensure all the parameters that affect the accuracy of the system could remain constant. Their system computed the error rate, in their case the error-rate was 20%. Eventually their goal is to have a system that works on a broad range of devices with less false acceptance rate as well as false rejection rate.

#### c. Other Validation Systems

Recent research investigated the possibility of determining whether the user was an imposter or not [7]. When determining verification accuracy, one must look at the FRR and the FAR. FRR, or the false rejection rate, is the probability that the user is wrongly identified as an imposter. The FAR, or false acceptance rate, is when the imposter is incorrectly identified as the user. In this research, it was found that there was a FAR 1.53. This is

too high of a false acceptance rate, the European Standard for Access Control Systems requires under a .001% false acceptance rate. The FRR was recorded at 5.65, again, the European Standard for Access Control Systems requires under a 1% false rejection rate.

Participants entered a nine digit numbered code. The participants would enter this code in with a mouse, into a, 0 – 9, keypad. Speed in a direction, and distance traveled were recorded. This study was done using ten undergraduate students, ages 22-25. Possible issues can arise with such a small sample size. It is not a good indication of the general population.

Another problem with this study is the testing method. The task chosen appears very simple. A task that is too simple could lead to a high FAR, which makes the verification method insecure. It seems highly likely that an imposter may have very similar behavioral movements for such short times and distances; also it would seem to be easy to mimic another user's patterns in such a task.

One additional test method for this task could provide an improvement. Perhaps if the angle in which the user moves the mouse was also tested, it would decrease the FAR and FRR. It would add a unique behavior to the task, making it more difficult for an imposter to mimic.

In a study by Zheng, Paloski, and Wang, research was done on how the mouse was moved versus where the mouse was moved [8]. User sessions were recorded, and the users' mouse data were recorded. Their study looked at the different angles and trajectories the users had during the user's session. The average FRR in this study was .86% and the average FAR was 2.96%. While these numbers are much better than Singh's study, it still does not meet the requirements for the European Standard for Access Control Systems.

### 3) MEASURING MOUSE BIOMETRICS

Mouse trajectories can arise from the following actions:

a. *System wake up* – the mouse is moved or jiggled to wake up the operating system (no mouse clicks at either end of the trajectory) [9].

b. *Move and click* – the mouse is moved to a location on the screen to perform an action such as clicking on an object, etc. The trajectory begins without a mouse event and ends in a mouse click.

c. *Highlight* – a section of text or an object is highlighted. This action begins with a left mouse click/hold to begin the highlighting and ends with the mouse release.

d. *Drag and drop* – an object is dragged and dropped. This action begins with a left mouse click/hold and ends with the mouse release.

The above categories combined with session level mouse trajectories produce 45 features shown in Table 1.

Action	Basic Feature Measurements	Sample For Each Feature
<b>System wake up:</b> The mouse is moved or jiggled to wake up the operating system (no mouse clicks at either end of the trajectory)	1. From the number of trajectory points 2. From the time of the trajectory 3. From the point-to-point distance 4. From the length of the trajectory 5. From the point-to-point velocities 6. From the point-to-point accelerations 7. From the point-to-point direction angle changes 8. From the number of inflection points 9. From the curviness of the trajectory	mean (average), median, minimum, maximum, standard deviation
<b>Move and click:</b> The mouse is moved to a location on the screen to perform an action such as clicking on an object, etc. The trajectory begins without a mouse event and ends in a mouse click.	1. From the number of trajectory points 2. From the time of the trajectory 3. From the point-to-point distance 4. From the length of the trajectory 5. From the point-to-point velocities 6. From the point-to-point accelerations 7. From the point-to-point direction angle changes 8. From the number of inflection points 9. From the curviness of the trajectory	mean (average), median, minimum, maximum, standard deviation
<b>Highlight:</b> A section of text or an object is highlighted. This action begins with a left mouse click/hold to begin the highlighting and ends with the mouse release.	1. From the number of trajectory points 2. From the time of the trajectory 3. From the point-to-point distance 4. From the length of the trajectory 5. From the point-to-point velocities 6. From the point-to-point accelerations 7. From the point-to-point direction angle changes 8. From the number of inflection points 9. From the curviness of the trajectory	mean (average), median, minimum, maximum, standard deviation

<b>Drag and drop:</b> An object is dragged and dropped. This action begins with a left mouse click/hold and ends with the mouse release.	1. From the number of trajectory points 2. From the time of the trajectory 3. From the point-to-point distance 4. From the length of the trajectory 5. From the point-to-point velocities 6. From the point-to-point accelerations 7. From the point-to-point direction angle changes 8. From the number of inflection points 9. From the curviness of the trajectory	mean (average), median, minimum, maximum, standard deviation
---	---	--

Table 1: *Session-Level Mouse Trajectory Features*

There are different formulas that can help in acquiring the data necessary to authenticate a user using mouse biometrics. The formulas below are used to get the wanted information.

a) *Amount of trajectory points*

$$\sum_{i=1}^n (P_i)$$

The number of points constitutes the entire trajectory; this formula determines the number of points involved in the entire trajectory [9].

b) *Amount of time to complete trajectory*

$$\sum_{i=2}^n (t_i - t_{i-1})$$

This formula measures the amount of time that the trajectory took to finish from the beginning until it reached the last point.

c) *Length of the trajectory*

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

This determines the length of the trajectory by adding all of the point to point distances, which is determined by the formula inside the square root.

d) *Velocity from point to point in the trajectory*

$$V_i = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}}$$

This formula takes into account the length of the trajectory from one point to another divided by the time

that it took to go from the previous point to the current point, so it calculates how rapidly it moved, which is the velocity from point to point.

e) *Acceleration from point to point in the trajectory*

$$a_i = \frac{V_i - V_{i-1}}{(t_i - t_{i-1})/2}$$

The velocity of the two points is divided by the length of time that it took to get from one point to the next gets the acceleration between the two points.

f) *Direction angle from point to point*

The formula to get the angle is

$$m_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

Then, to get the change in angle between the two points, the formula is

$$m_i - m_{i-1}$$

### 3.1 Mouse Click Characteristics

A mouse click takes place when a button of the mouse or the scroll wheel in the middle is pressed and then released [9]. The five types of mouse click events: left click, right click, double click, highlight, and drag and drop. The highlight and drag-and-drop events are combined and are referred here as drag-and-drop events. Table 2 displays the 29 features that were implemented.

Mouse Click Event	Mouse Click Features
Ratio of left clicks Ratio of right clicks Ratio of double clicks Ratio of drag-and-drop clicks	To the total number of mouse clicks
Average number of mouse clicks Average number of left clicks Average number of right clicks Average number of double clicks Average number of drag-and-drop	Clicks per minute
Mean dwell time Median dwell time Minimum dwell time Maximum dwell time	Of all left or right mouse clicks
Mean dwell time Median dwell time Minimum dwell time Maximum dwell time	Of all left mouse clicks
Mean dwell time Median dwell time Minimum dwell time Maximum dwell time	Of all right mouse clicks

Mean transition time Median transition time Minimum transition time Maximum transition time	Of all double clicks
Mean transition time Median transition time Minimum transition time Maximum transition time	Of all drag-and-drop clicks

Table 2: *Mouse Click Features*

### 3.2 Mouse Scroll Characteristics

Mouse scrolling refers to the turning of the wheel in between the two buttons of the mouse. It can scroll up or down while making the application move in that direction respectively.

The following 40 features shown in table 3 were implemented [9]:

Mouse Wheel Spin /Scroll Event	Mouse Wheel Spin/ Scroll Features
Ratio of scroll up Ratio of scroll down	To total number of wheel events
Ratio of time spent in wheel events	To the total sample session time
Mean duration in seconds Median duration in seconds Minimum duration in seconds Maximum duration in seconds	For a wheel spin event
Mean duration in seconds Median duration in seconds Minimum duration in seconds Maximum duration in seconds	For a scroll up event
Mean duration in seconds Median duration in seconds Minimum duration in seconds Maximum duration in seconds	For a scroll down event
Mean scrolled distance Median scrolled distance Minimum scrolled distance Maximum scrolled distance	For a scroll up or down event
Mean scrolled distance Median scrolled distance Minimum scrolled distance Maximum scrolled distance	For a scroll up event
Mean scrolled distance Median scrolled distance Minimum scrolled distance Maximum scrolled distance	For a scroll down event
Mean speed (speed=distance/time) Median speed Minimum speed Maximum speed	Of wheel spin events
Mean speed (speed=distance/time) Median speed Minimum speed Maximum speed	Of scroll up events
Mean speed (speed=distance/time) Median speed Minimum speed Maximum speed	Of scroll down events
Ratio	Of scroll up to scroll down events (default zero)

Table 3: *Mouse Wheel Spin/ Scroll Features*

## 4) MOUSE BIOMETRIC DATA

In order for the key-logger to be running, the user has to go to the website and log into his/her account or register if they have not signed up before. The user can pick 'Free Input' from the drop-down menu and click on the launch button to start.



Fig. 4. Webpage where key-logger app can be launched after logging in.

A file is downloaded and opened with Java Web Start. When this file runs it opens the key-logger and data can be seen being outputted by the application any time the mouse cursor moves or the buttons are clicked.

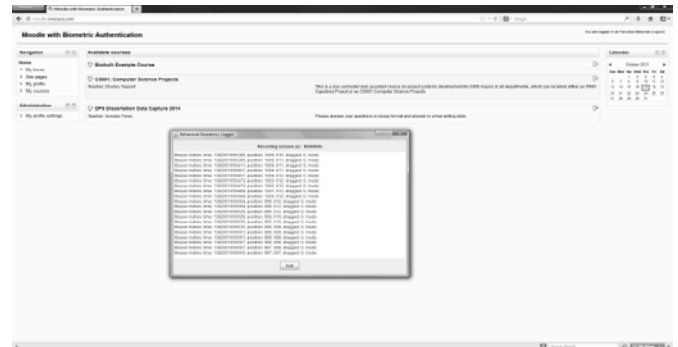


Fig. 5. Key-logger outputting mouse data as the mouse moves or the buttons are clicked.

While the application is running, the bubble application can be opened and the data from the user clicking on the bubbles will be outputted to the key-logger application. After the application has attained enough data from the session, the key-logger application is ended and the data will be uploaded into the database.

## 5) TESTING THE MOUSE

### a) Bubble Click Application

We have an application in mind that would make a bubble appear in a random spot on the screen, and the user is supposed to click on the bubble. Some bubbles need to be clicked once and some distinguishable bubbles need to be clicked on twice. The bubble will keep disappearing as the user clicks on them until the maximum amount of

bubbles has been reached, which is 20 bubbles. From the 20 bubbles that the user will encounter, five of them will have an inner shaded circle; these have to be double-clicked by the user in order for the next bubble to appear.

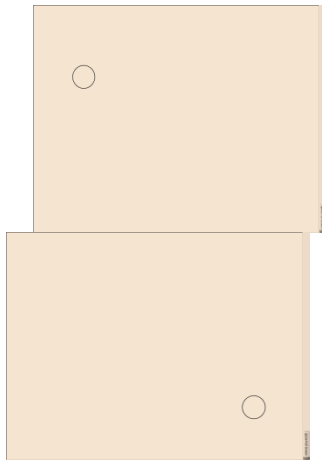


Fig. 6. Bubble comes out in a random part of screen, and after being clicked appears in another part of the screen.

As mentioned before, some of the bubbles will be noticeably different from the rest and these have to be double-clicked as opposed to being single clicked. They will have an inner shaded circle that will be shaded in. These are to test the user's double-click speed as well as the velocity and the distance that the mouse traveled. Is it yet another test to determine authentication accurately.

The randomness of where the bubbles appear depends on a random generator that outputs a certain location on the screen. The random generator outputs a height and a width output and these numbers are used to tell the bubble where on the screen to appear. The random generator also takes into consideration the timestamp, so it is highly unlikely that a bubble will appear in the same exact spot as the previous one.

#### *b) Bubble Data Gathering*

Once the user has finished clicking on all of the bubbles and the key-logger has ended collecting the data, the data will be uploaded into the database where once we are granted access we can experiment and run tests on the data using the previous formulas for determining velocity, distance, acceleration, and angle change and can determine patterns or certain traits that we can use to authenticate the user.

The bubble application works by measuring a person's acceleration and velocity once a bubble appears on the screen, it measures the user's click speed when the user clicks on the bubble. The velocity and acceleration are once again tested when the clicked bubble disappears and a new one appears on the screen. The double-click speed of the user is tested when they encounter a bubble with a shaded inner circle. The double-clicked bubbles are to ensure authentication because certain users can have the same single-click speed but by adding a double-click test, the chances of it being multiple users greatly diminishes.

#### *c) User Authentication*

A user's authentication can be determined by quantifying the data into measurable chunks that can be used to determine specific characteristics and traits. The user data is compared to logged data, and a match can be determined depending on the similarities between the unknown user and the data on the database. If a match is confirmed based on the velocity, acceleration, distance, and single and double-clicked speeds the user is authenticated and access is granted to the user. This is certainly another viable method for user authentication through a computer than just the common methods that we have seen in the past.

#### *d) Similarities in Applications*

The bubble application has some fun aspects about it because the user is expecting a bubble to appear but they do not know where it will appear. The bubble application has some similarities to some well-known games, which makes it a good method for user authentication. The fact that the user will be actively engaged in the biometric instead of doing it without any energy or desire means that they will not get skewed results. The bubble application is similar to solitaire in the sense that the user can click on the deck to get the next card, the user can drag and drop cards, the user can double click items to put them on the stacks and the user has to move the mouse to different parts of the screen before time runs out.

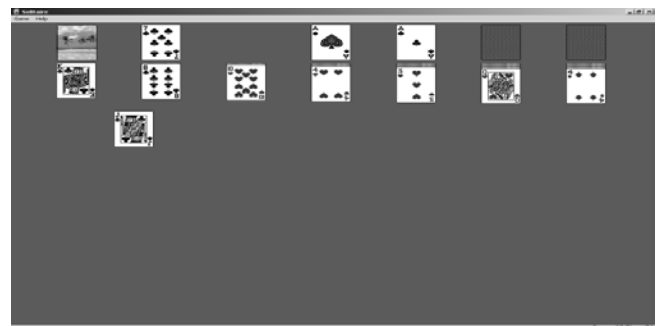


Fig. 7. Solitaire application showing the mouse clicking and dragging a card to another stack.

## **6) FUTURE STUDY**

While mouse dynamics offers a mostly effective authentication method in certain environments, it may suffer from rather glaring issues that prevent its use on an end-user basis. One of the most problematic issues is the length of time that verification requires in such implementations. Although studies mention successful verification statistics, the vast majority of such studies fail to identify the exact length of the times used by such neural networks. In further analysis, it was determined that very few of the systems had an implementation time of under 2%, which highly restricts their deploy ability and marketability to the consumer and commercial markets. Users are highly picky toward the wait times with the use of verification technologies, and a wait time of over a

minute would render such a system ineffective as there would be very few advantages if it prevents a natural computing workflow from occurring. While this may be negated through software innovation and different modeling techniques, it does not remove the reality that such methods require statistical computation that is tedious to effectively integrate and implement.

Furthermore, such systems cannot fully control various variables that exist within a working environment, which may significantly increase the rate of false positives in such systems. Users might be using different mouse devices or different hardware and the results can be influenced by those variables, and the authentication might not be accurate [6]. Changes in workflows through different machines and tasks can render such analysis moot as different working environments create a shift in behavior within the individual. This unpredictability will not be able to be accounted for by the behavioral biometric analysis techniques, and would serve as an ineffective identification of the user. As such, innovations such as remote desktop use and the use of other input technologies would render this use moot.

## 7) CONCLUSION

While initially promising, the implementation of mouse dynamics suffers from various theoretical and physical flaws that leave such a system as unusable and counterproductive in a workplace scenario. Even when the system has demonstrated a significant level of success in identifying the user of a system, it is either slow, inflexible to meet worker demand or is worthless as a stand-alone verification system due to built-in weaknesses. As such, even a fully implemented system would function best as a complement to a much more effective system rather than as the primary system itself. The fusion of dual biometric systems might seem like a more viable method for authentication [10]. Combining mouse biometrics would yield more accurate results, and it would be a seamless transition for the users because other peripherals like the keyboard have to be used for normal computer use [6].

Ultimately, while research has been inconclusive toward the role that mouse dynamics will play in verification techniques, there is significant doubt toward its implementation and general viability as a biometrical system. Due to the shift in input methods in recent times, such a system may be wholly outdated.

## REFERENCES

- [1] C. Brodley, M. Pusara, "User Re-Authentication via Mouse Movements". ACM workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC '04), 2004.  
Available: <http://www.csis.pace.edu/~ctappert/it691-11fall/projects/mouse-pusara.pdf>
- [2] Wikipedia contributors. (2013, October 16). *Mouse (computing)* [Online].  
Available:  
[http://en.wikipedia.org/wiki/Mouse\\_%28computing%29](http://en.wikipedia.org/wiki/Mouse_%28computing%29)

- [3] Wikipedia contributors. (2013, October 16). *Biometrics* [Online].  
Available: <http://en.wikipedia.org/wiki/Biometrics>
- [4] R. Kaminsky, M. Enev, E. Andersen, "Identifying Game Players with Mouse Biometrics". University of Washington, Seattle, WA, 2008.  
Available:  
[http://homes.cs.washington.edu/~miro/docs/mouse\\_ID.pdf](http://homes.cs.washington.edu/~miro/docs/mouse_ID.pdf)
- [5] S. Hashia, C. Pollett, M. Stamp, "On Using Mouse Movements as a Biometric". San Jose State University, San Jose, CA, 2005.  
Available: <http://www.cs.sjsu.edu/faculty/pollett/masters/Semesters/Spring04/Shivani/shivanipaper.pdf>
- [6] Z. Jorgensen, T. Yu, "On Mouse Dynamics as a Behavioral Biometric for Authentication". North Carolina State University, Raleigh, NC, 2011.  
Available: <http://www4.ncsu.edu/~tyu/pubs/asiaccs11-jorgensen.pdf>
- [7] S. Singh, K. Arya, "Mouse Interaction based Authentication System by Classifying the Distance Travelled by the Mouse". *International Journal of Computer Applications*, vol. 17, pp. 45–47, March 2011.  
Available: <http://www.ijcaonline.org/volume17/number1/pxc3872752.pdf>
- [8] N. Zheng, A. Paloski, H. Wang, "An Efficient User Verification System via Mouse Movements". The College of William and Mary, Williamsburg, VA, 2011.  
Available: <http://www.cs.wm.edu/~hnw/paper/ccs11.pdf>
- [9] P. Xavier de Oliveira, V. Channarayappa, E. O'Donnell, B. Sinha, A. Vadakkencherry, T. Londhe, U. Gatkal, N. Bakelman, J.V. Monaco, and C.C. Tappert, "Mouse Movement Biometric System," *Proc. CSIS Research Day*, Pace University, NY, May 2013.  
Available: <http://csis.pace.edu/~ctappert/srd2013/b2.pdf>
- [10] A. Ross, A. Jain, "Information fusion in biometrics". Michigan State University, East Lansing, MI, 2003.  
Available: [http://www.cse.msu.edu/rgroups/biometrics/Publications/Multibiometrics/RossJain\\_BiometricFusion\\_PRL03.pdf](http://www.cse.msu.edu/rgroups/biometrics/Publications/Multibiometrics/RossJain_BiometricFusion_PRL03.pdf)