Authentication by Mouse Movements

By

Shivani Hashia

Advisor: Dr. Chris Pollett

Committee: Dr. Mark Stamp

Dr. Robert Chun

Dec 2004

Topics

- Introduction
- Design and Implementation
  - Active authentication
  - Passive authentication
- Experiments and Results
- Conclusion
- Future Work

Introduction

- Authentication: Process of validating a person is who he claims to be
- Aim:
  - Build a secure technique to authenticate users
  - Easy to use
  - Cheap

Current methods of authentication

- Passwords- Based on what you know
- **Advantages**
  - Easy to use
  - Cheap
- **Disadvantages**
  - Difficult to remember
  - Can be cracked if not chosen wisely

Current methods of authentication (cont'd)

- Smart Card- Based on what you have
- **Advantages**
  - Easy to use
- **Disadvantages**
  - Expensive, requires card reader
  - Can be stolen

Current methods of authentication (cont'd)

- Biometrics- Based on physiological or biological characteristics
- **Advantages**
  - No one can forge unless template file changed
- **Disadvantages**
  - Current methods expensive

Related work

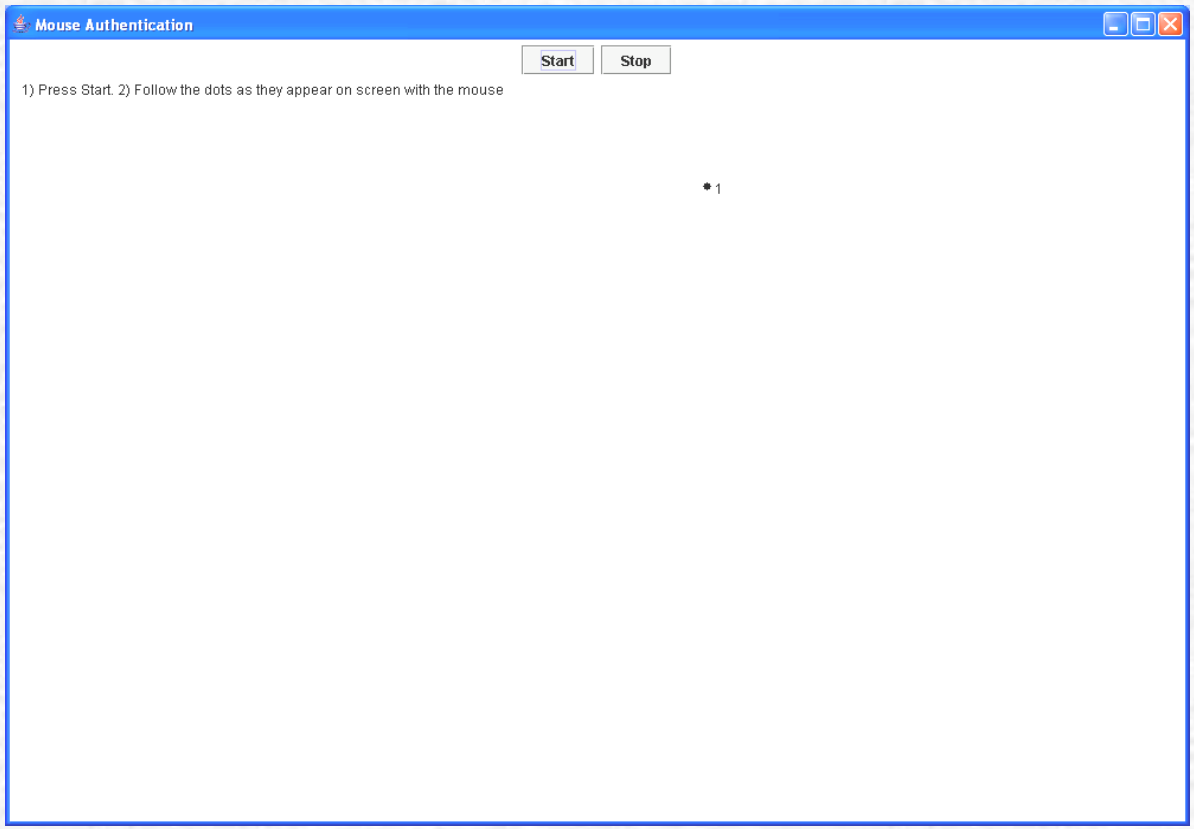
- Researchers at Technion-Israel Institute of Technology were planning to built software, which can identify the authenticity of the users with their individual and distinct typing styles

- Ross Everitt and Peter McOwan at Queen Mary University of London did a research where they used mouse signature as the biometric to verify the authenticity of the users

Design and Implementation

- Model based on biometrics. It contains two parts:
  - **Active Authentication:** One time authentication
  - **Passive Authentication:** Continuous monitoring and authentication of mouse movements

Active Authentication

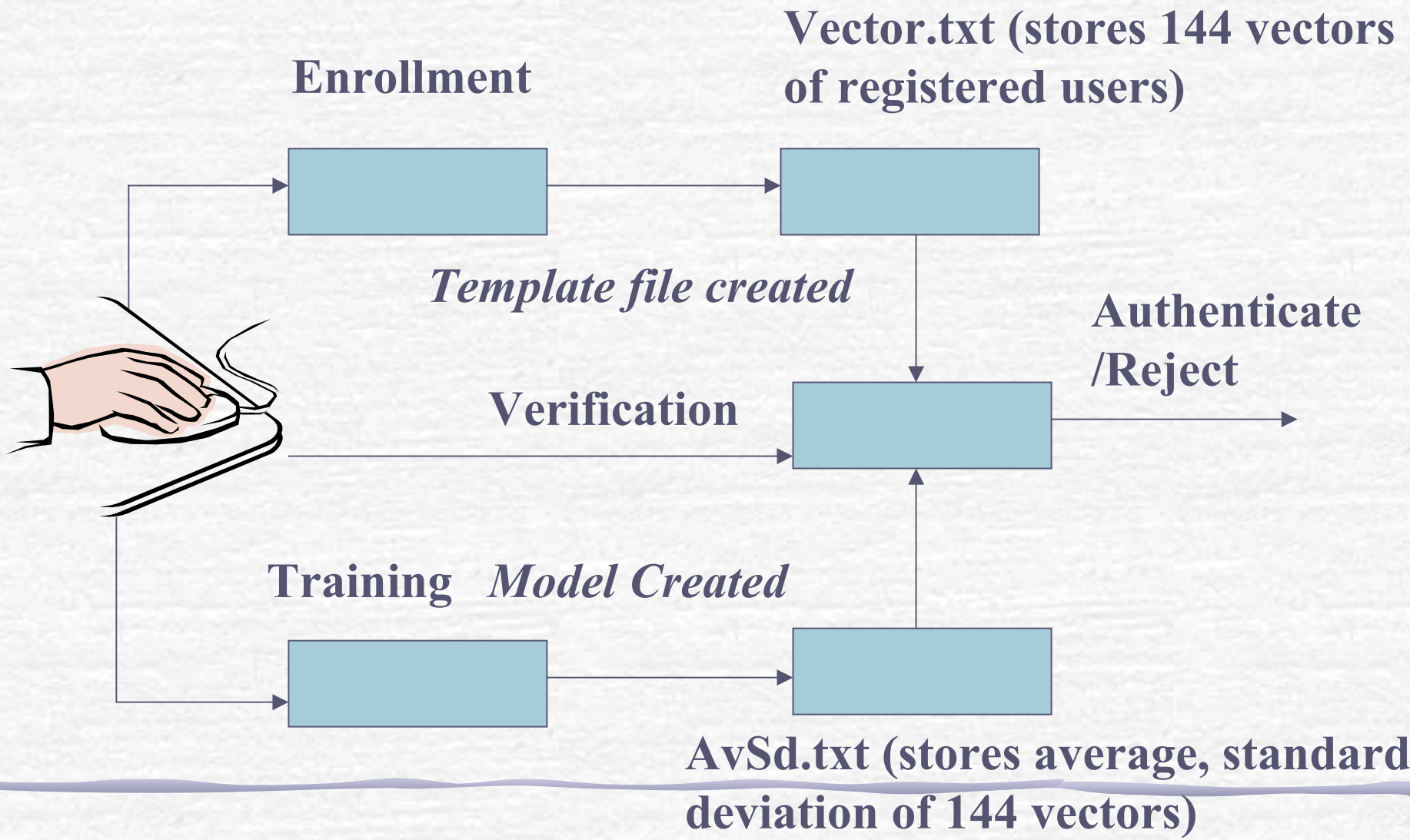


Login Screen

Active Authentication

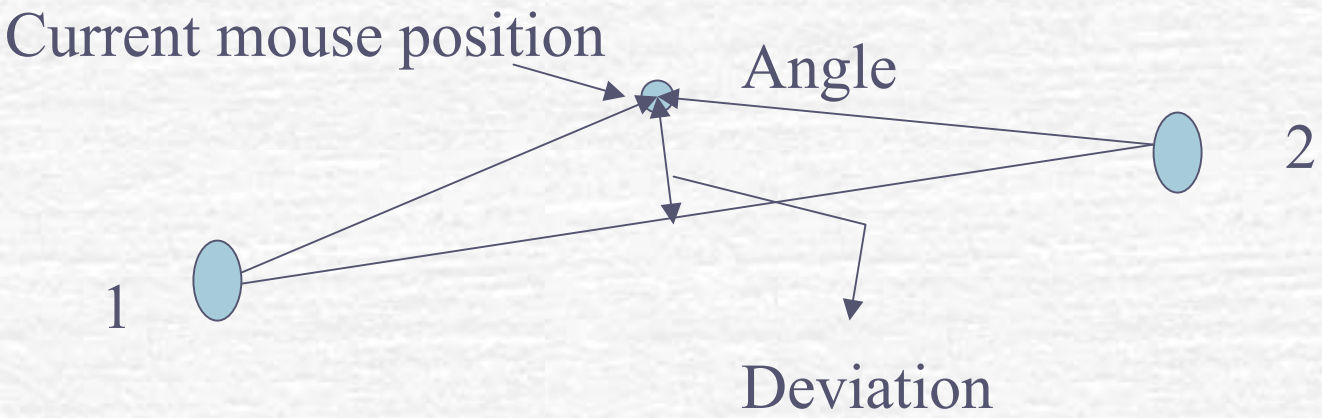
- Works in 3 phases
  - **Enrollment:** For registering users. Acts like mean of data points
  - **Training:** For getting as many samples as possible from users. Acts like variance of data points
  - **Verification:** Actual logging in

Design


Enrollment
Vector.txt (stores 144 vectors of registered users)
Template file created
Authenticate /Reject
Verification
Training
Model Created
AvSd.txt (stores average, standard deviation of 144 vectors)

Enrollment

- Complete round of following points on screen with mouse four times
- Record parameters- speed, deviation, angle (positive , negative)



Enrollment (cont'd)

- Find average, standard deviation, maximum and minimum of the four parameters
- For 1 pair of points, 4*4=16 vectors
- For 9 pair of points, 16*9=144 vectors
- Normalize vectors to bring on same scale
- Store vectors in file
- It acts like mean of vectors

Training

- Complete rounds 20 times
- Get variation in user's mouse movements
- Repeat procedure as during enrollment
- Store 144 vectors in a temporary file
- Acts like variance of vectors

Training (cont'd)

- Find difference between each training phase data set and corresponding vectors in enrollment
- Average the differences of each vector
- Find standard deviation of differences for each vector
- Store average, standard deviation for each vector difference in a file

Training (cont'd)

Enrollment Vectors

r 0.2955206224054584 6 8.314035619802 09 172.5627504337201

Subtract **Training Vectors**

r 0.3138153686526 77 8.386129099100515 154.35863930928812
r 0.3128763690714 65 5.112780770424284 159.4521907242159
r 0.3465478372758716 4.271812610295602 178.0942234512897

Store average, standard deviation of differences

Verification

- User given login screen
- Has to move mouse on the screen
- Speed, deviation, angle calculated
- 144 vectors from average, standard deviation, maximum, minimum calculated
- Vectors normalized

Verification (cont'd)

- Find difference of verification vectors with corresponding vectors during enrollment
- Check if each difference lies in its corresponding range of average-1.5*standard deviation and average+1.5*standard deviation

Verification (cont'd)

- Count the number of vectors that lie in their defined range for each user
- Repeat same for training phase data to get range of counters for the user trying to verify
- If count falls within the range for that user and is the greatest for the user, he is authenticated

Verification(cont'd)

Enrollment Vectors(vector.txt)

r 0.29552062240545846 8.31403561980209 172.5627504337201

- *Subtract*

Verification Vectors (tmpVector.txt)

r 0.28212139796611 69 3.07889832001817 5 178.2263299758116

Average, standard deviation (AvSd.txt)

r 0.212132278976185 1.2734653494142 -2.47539960382224 9

Check if within range of average+- 1.5*standard deviation

For vector 1, range =0.212132278976185+-1.5*1.2734653494142

Passive Authentication

- Idea to keep eye on user's movements
- Runs in background

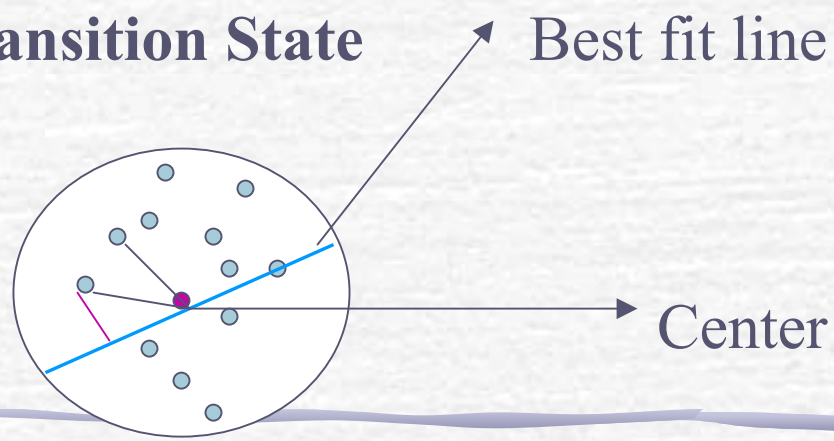
- Two phases :
  - Enrollment
  - Verification

Enrollment

- Record mouse movements for 15 minutes
- Find dense regions on screen
- Draw convex hulls around dense regions
- Treat hulls as transition states

Enrollment (cont'd)

- For transitions within same state calculate speed, distribution of points around the center, distance from best fit line, frequency



- Mouse position
- Center of hull

Enrollment (cont'd)

- For each state calculate average of speed, angular distribution and distance from best fit line
- Also find standard deviation of speed, angular distribution and distance from best fit line
- Store data in file

Verification

- Record movements continuously
- After every 2 minutes, calculate speed, angular distribution and distance from best fit line
- Check if they lie in the range for average-1.5*standard deviation to average +1.5*standard deviation for respective parameters

Verification(cont'd)

- If majority of the data points lie within the specified range, keep on continuing
- Update files

Experiments and Results

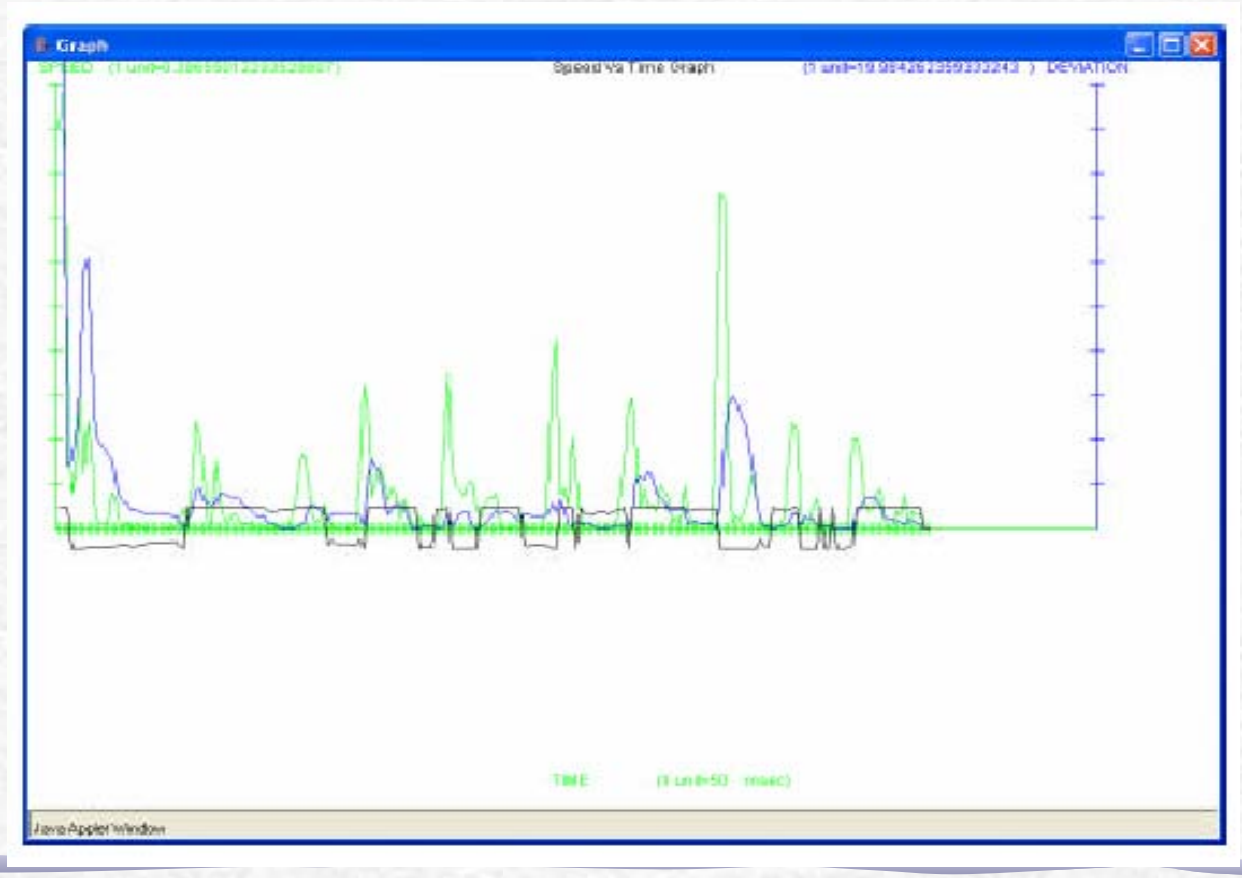
- Performed a number of experiments to find a way to use mouse movements as authentication method
- Active Authentication
  - Find parameters unique to users
  - Find ways to use the parameters so that users are authenticated

Experiments (cont'd)

- Passive Authentication
  - Find way to record mouse coordinates in background
  - Filter recorded data coordinates
  - Draw bounded regions around dense regions to form transition states
  - Find parameters and a way to use them

Active authentication

- Find parameters unique to users



Active authentication

- Ways to use the parameters
  1) Check if the sum of square of differences is within a certain threshold
  2) Use the lowest sum of square of differences

  - Result: FAR of 40%, FRR of 40%

FAR= False Acceptance Rate – Forged user, system accepts (fraud rate)

FRR= False Rejection Rate – Actual user, system rejects (insult rate)

Active authentication

3) Select specific vectors

- Calculate maximum and minimum sum of square of differences between user's registered and training phase vectors
- Use lowest sum of square of differences and check if falls within range
  - Result: FRR =12%, FAR=70%

Active authentication

4) Using specific vectors in different ways to get common set of vectors
   - Compute sum of square of differences
   - Check for the lowest sum of square of differences and see if lies in the range of differences

Active authentication

- Results

	All parameters	Specific parameters	Union of specific parameters	Intersection of specific parameters
FRR	65%	12%	50%	57%
FAR	23%	70%	47%	57%

Active authentication

5) Using difference, average, standard deviation for specific vectors
- Calculate difference between verification and registered vectors
- Check if they fall within the range of average+-standard deviation
- Count number of vectors that follow the criterion

Active authentication

Results

	Specific parameters Av+3*sd	Specific parameters Av+2*sd
FRR	40%	70%
FAR	36%	15%

Active authentication

6) Using difference, average, standard deviation for all vectors

- Count number of vectors for which difference between verification and registered vectors lies between average-1.5*standard deviation to average+1.5*standard
- Repeat for every user
- Select one with the highest count

Active authentication

7) For each attempt of training phase, we also found the number of vectors which fall within the defined range

Found range of counters for individual users from training phase

- Each user has his own range
- During verification, checked if the count of vectors is within the range specified

Active Authentication

Using individual ranges. Separate model for everyone

Model 1	FAR	FRR
1 user	20%	0%
3 users	31.5%	0%

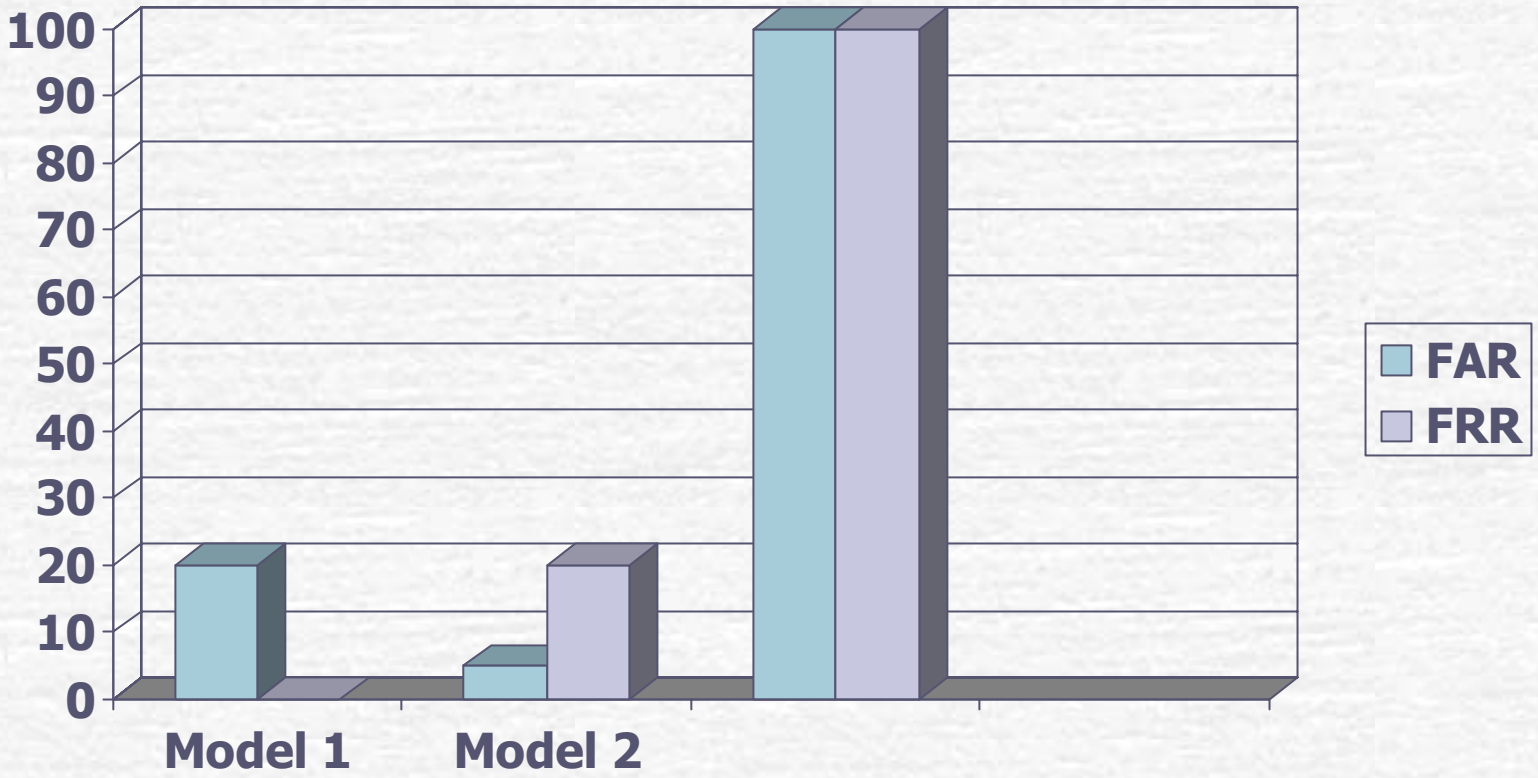
Active authentication

Combining individual ranges and comparison with other users

Model 2	FAR	FRR
1 user	5%	20%
3 users	13.1%	25%

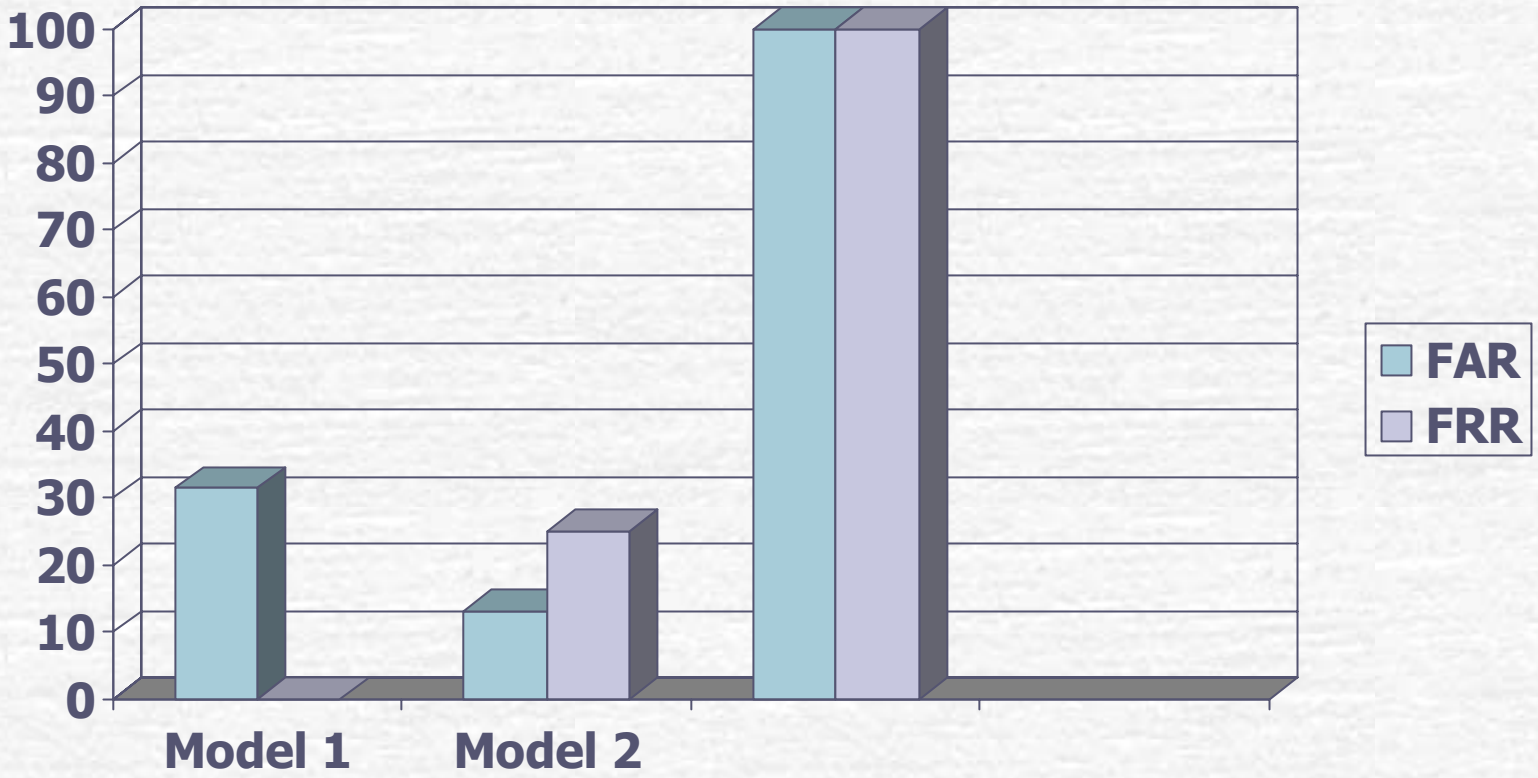
Active authentication

For single user



Active authentication

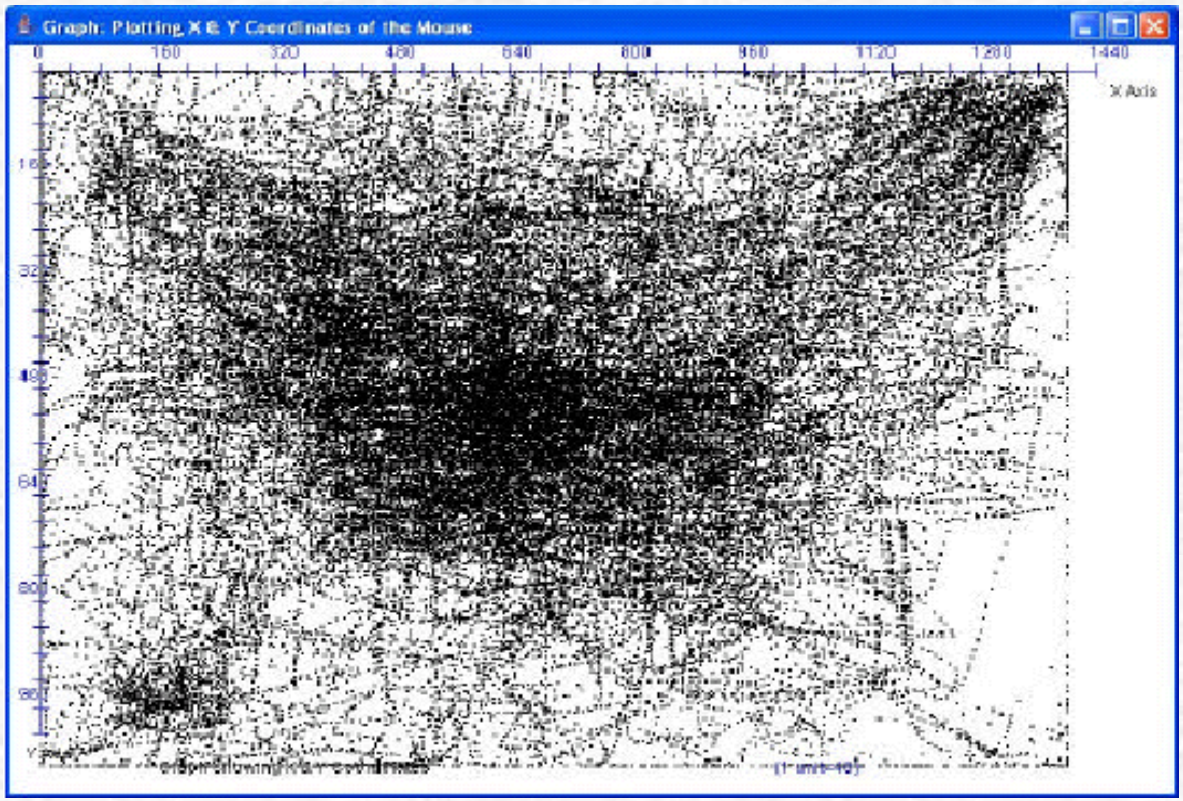
For 3 users



Passive authentication

1) Record coordinates in background
   - Used Windows hooks
   - Recorded data for about 4 hours

Passive authentication

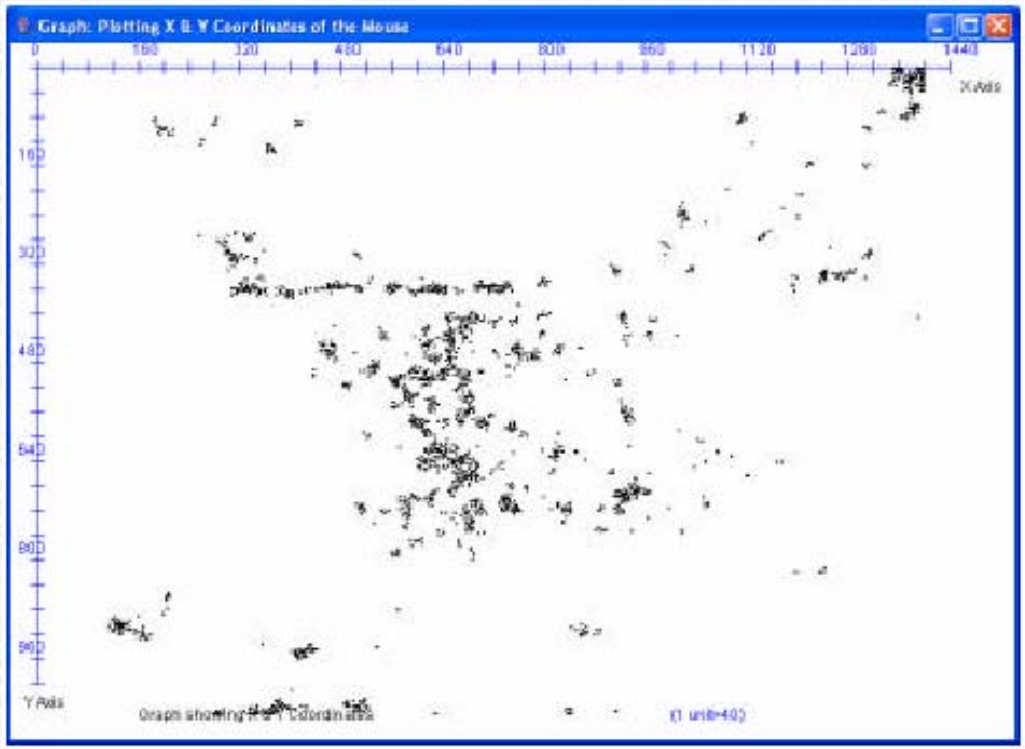


Passive authentication

2) Filter data points
    1) Delete coordinates just before and after mouse went idle
    2) Delete coordinates where the speed was above a threshold
- Result: Did not make a significant change in the concentration of points

Passive authentication

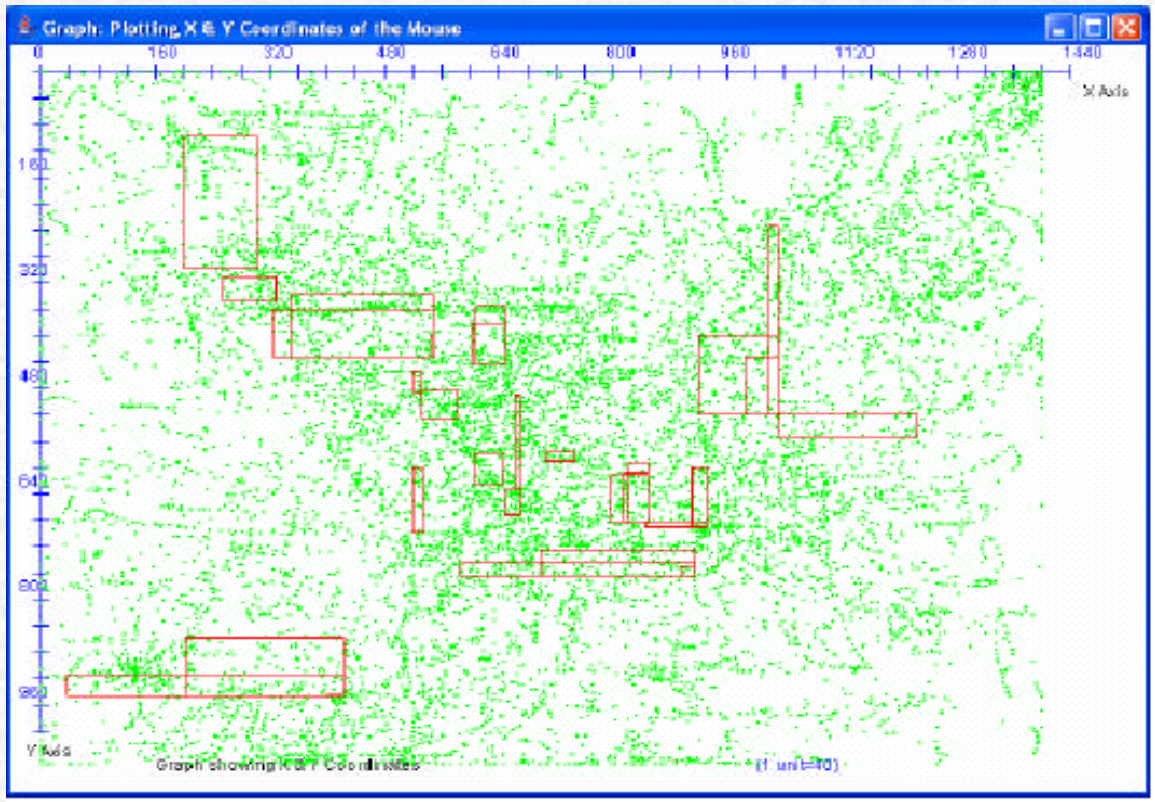
3) Selecting points around which density of points was above a certain number



Passive authentication

4) Enclose dense regions with a bounded figure
   - Draw a rectangular region around dense regions
- Result: Couldn’t get all the dense regions

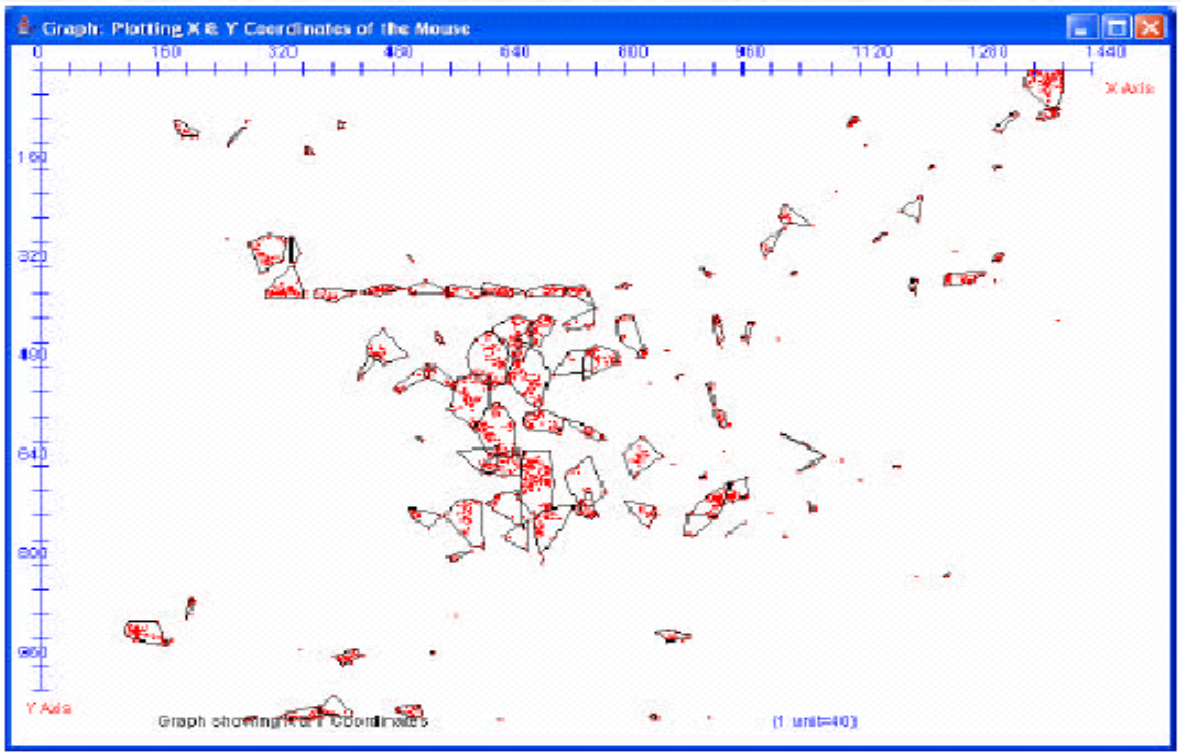
Passive authentication



Passive authentication

- Using gift-wrapping algorithm

- Result



Passive authentication

5) Find vectors for authentication

- Checked for transitions between states
- Limited transitions within same state
- Recorded speed, angle, distance from best fit line
- Stored average, standard deviation of three parameters

Passive authentication

- Results
  - FAR=90%
  - Hopeful it would work if add more parameters to differentiate users

Conclusion

- Security important in today's world
- Need security systems that are cheap and easy to use
- Authentication by mouse movements provides both

Conclusion (cont'd)

- Active authentication has FAR and FRR of 13% and 25 % respectively
- FAR and FRR increase if users have overlapping regions of similarity
- Passive authentication can be improved by adding more parameters

Future Work

- Need to reduce FAR and FRR
- Can add some more parameters to make it more precise
- Make improvements in passive authentication

Q&A

Demo

Thank you