SlowMo – Enhancing Mobile Gesture-Based Authentication Schemes via Sampling Rate Optimization

Kent W. Nixon, Xiang Chen, Zhi-Hong Mao, Yiran Chen
University of Pittsburgh
January 27, 2016
Outline

• Current State of Mobile Security
• Our Gesture-Based Mobile Security Solution
• Data Capture and Pre-Processing
• Results and Analysis
• Conclusion
Proliferating Devices

• Over 26 billion smart connected devices before 2020
• Many household objects embedded with electronics and sensors
Data Stored On Smart Devices

• 2/3 of people store a moderate or a significant amount of personal data on their smart mobile devices

• 1/3 of mobile phones are misplaced at least once a year

Percent of finders of “lost” phones who accessed...

- Contacts: 81%
- Cloud-Based Docs: 47%
- Social Networking: 64%

- Passwords: 57%
- Salary Info: 45%
- Online Banking: 43%
Existing Security Methods

• Robust security methods become of increasing importance
• Most popular schemes like PIN and pattern lock remain easily bypassed
Existing Security Solutions

- Emerging trend is integration of fingerprint scanners
- Android devices allow for facial recognition
- Unique and difficult to reproduce, yet user can be forced to supply information
Smart Device Security

- 51% have neither keypad locks nor passwords on their smartphone.
- 43% of consumers consider security features of importance.
- 19% have passwords.
- 10% have both keypad lock and passwords.
Ethological Security

• Use of user activity as a form of authentication

• Most promising is gesture-based security
  • The way in which a user interacts with a device as authentication
What’s In a Gesture

• “Gesture” is a repeatable, recordable pattern created when a user interacts normally with a device

• Examples: the way a user holds a device, how they interface with the screen, daily routines/activities
Benefits of Gesture-Based Security

• Functions using only existing sensors
  • No new hardware, drivers, OS support, etc.

• Difficult to hack
  • Degrees of freedom are large and brute-force attacks are all but impossible

• Constantly changing identifier
  • Usage patterns slowly change over time

• Protects the user
  • Injuring the user could make the device useless to everyone
Is A Gesture Unique?

• In a word: yes
• Everyone has their own unique physical make up

The hand alone has 29 pieces of bone, 123 pieces of ligament, 35 pieces of muscle, and 48 strips of nerves
Is A Gesture Unique?

• Everyone has a preferred way to interact with a device

The video to the left shows two users entering the same PIN on the same phone. They were given no instructions other than to simply enter the code.
Are Gestures Repeated?

• It is not enough that gestures are unique
• Must be repeatable
• Verify this experimentally

To the left are screenshots from within an Android app we created to record users’ gestures. Shown are the test cases where the user is asked to enter a text message and when the user is asked to swipe.
We compared three volunteers’ Pressing Time Distributions (PTD) for the Password of 1245. Not only is the PTD unique per user, there is very little variation between samples.
PIN Press Timing Distributions

- Across different PIN’s, user PTD remains the same
- Indicates that at least some gesture features are constant for a single user
Further Analysis

• Significant difference between users
• Only small deviation across same user
• Verifies that gestures are unique and repeated

<table>
<thead>
<tr>
<th></th>
<th>Avg. Contact Area</th>
<th></th>
<th></th>
<th></th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9043</td>
<td>1245</td>
<td>3179</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-Age Female</td>
<td>0.1707</td>
<td>0.1762</td>
<td>0.1715</td>
<td>0.0029</td>
<td></td>
</tr>
<tr>
<td>Young Female</td>
<td>0.1456</td>
<td>0.1541</td>
<td>0.1529</td>
<td>0.0046</td>
<td></td>
</tr>
<tr>
<td>Young Male</td>
<td>0.1977</td>
<td>0.2008</td>
<td>0.1998</td>
<td>0.0015</td>
<td></td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.0212</td>
<td>0.0191</td>
<td>0.0192</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Avg. Pressure</th>
<th></th>
<th></th>
<th></th>
<th>Std. Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9043</td>
<td>1245</td>
<td>3179</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-Age Female</td>
<td>0.5293</td>
<td>0.5561</td>
<td>0.5411</td>
<td>0.0134</td>
<td></td>
</tr>
<tr>
<td>Young Female</td>
<td>0.4611</td>
<td>0.5013</td>
<td>0.4910</td>
<td>0.0208</td>
<td></td>
</tr>
<tr>
<td>Young Male</td>
<td>0.5668</td>
<td>0.6100</td>
<td>0.5796</td>
<td>0.0222</td>
<td></td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.0436</td>
<td>0.0443</td>
<td>0.0363</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Record Gestures

• The Android operating system allows access to all sensors on a device
• Includes GPS, magnetometer, barometer, light sensor, proximity sensor, touchscreen, accelerometer, gyroscope, and many more
• With data from all of these sensors, we can easily capture gesture data from a user
Representing Gesture

- 11 features across 3 sensors
- Single gesture can be visualized as a set of sample points

Touch Screen
- Time Stamp
- Touch x-Location
- Touch y-Location
- Touch Area
- Touch Pressure

Accelerometer
- x-Axis Acceleration
- y-Axis Acceleration
- z-Axis Acceleration

Gyroscope
- x-Axis Orientation
- y-Axis Orientation
- z-Axis Orientation
Capturing Gesture

• Created an Android application that mimics the pattern lock
• Asked all users to enter the same pattern
  • Eliminates possibility of identifying user through differences in the patterns themselves
Android and Sensors

• Sensor sampling is non-deterministic
  • “It’s ready when it’s ready” attitude
• Sample rate fluctuates depending on a number of factors
• Gestures also last varying amounts of time
• Different sensors poll at different rates
• Most machine-learning algorithms expect data vectors of a set size
Standardizing Samples

- Gestures varied between 0.4 and 1.4 seconds
- Recorded anywhere between 24 and 162 samples per gesture
- Also differs depending on sensor
Standardizing Data Size

• Two ways to standardize the data
  • “Temporal” Standardization
    • Set gesture time, equidistant sample points
  • “Spatial” Standardization
    • Allow unique time per gesture, equal number of samples for each
Interpolating Data

• Discrete sample points do not always occur where needed (temporally) for standardization

• Perform linear interpolation to estimate the sample data at the required points

\[ y_{nm}^{(i)} = \frac{t_{x,n_v}^{(i)} - t_{y,n_m}^{(i)}}{t_{x,n_v}^{(i)} - t_{x,n_u}^{(i)}} x_{n_u}^{(i)} + \frac{t_{y,n_m}^{(i)} - t_{x,n_u}^{(i)}}{t_{x,n_v}^{(i)} - t_{x,n_u}^{(i)}} x_{n_v}^{(i)} \]
Normalizing Sensor Data

- Raw sensor values vary in magnitude
- Can artificially weight specific sensor values
- Must “normalize” all data to similar ranges
  - Naïve Normalization
  - Z-Score Normalization
- Examine impact of these, as well as without
SloMo Functionality

- **User & Device Interaction**
- **Touch Screen & Other Sensors**
- **Pattern Rec. & Data Collection**
- **Continuous Verification & Authorization**
- **Data Base & Gesture Behavior Bio-Modeling**
- **Application with Specific Gesture Behavior Pattern**
- **Net Computation & Storage Support**

1. **Input**
2. **Monitor**
3. **Real-time Pattern**
4. **Continuous Verification & Authorization**
5. **Application Run-time**
6. **Security Control**
7. **Security Activities**

- **Pattern Detection**
- **Pattern Discovery**
- **Verify & Build New Gesture Pattern**
- **Build-in Verification**
- **Data Support**
- **Data Transaction**
- **Support**
- **Security Control**

**Activities**
- **Continuous Verification & Authorization**
- **Net Computation & Storage Support**
- **Data Base & Gesture Behavior Bio-Modeling**
- **Application with Specific Gesture Behavior Pattern**

**Data Base & Gesture Behavior Bio-Modeling**
- **Data Transaction**
- **Support**
- **Build-in Verification**
- **Data Support**
- **Continuous Verification & Authorization**
- **Application Run-time**
- **Security Control**
- **Security Activities**

**Continuous Verification & Authorization**
- **Specific Pattern**
- **Operation Obj.**
- **Application Run-time**
- **Security Control**
- **Security Activities**

**Data Base & Gesture Behavior Bio-Modeling**
- **Data Transaction**
- **Support**
- **Build-in Verification**
- **Data Support**
- **Continuous Verification & Authorization**
- **Application Run-time**
- **Security Control**
- **Security Activities**
Types of Classification

• One Versus Many (OVM)
  • Identifying a single user out of a group of known users
  • Imagine an iPad which could determine which family member was using it at any given time

• One Versus All (OVA)
  • Identifying whether or not a user is the owner of the device
  • No a priori sample sets for other users
  • Imagine a smartphone that could be used by you but no one else
OVM

- Have sample data from all possible users
- Generate a Gaussian model for each user

Fig. 3: Accuracy results using a multivariate Gaussian classifier. The top row is spatially standardized, the bottom row is temporally standardized. From left to right: no normalization, naïve normalization, and z-score normalization.
OVA

• Only sample data from a single user
• Use a threshold for Gaussian classifier

![Equal error rates utilizing a multivariate Gaussian classifier. The top row is spatially standardized, the bottom row is temporally standardized. From left to right: no normalization, naïve normalization, and z-score normalization.](image)
Takeaways

• Temporal standardization
• No more than 20 training samples required
• Max accuracy achieved with low sample rate
Conclusion

• Shows different classification methods have different data processing requirements
• Indicates ethological information that is shared between all users
• Demonstrates that low sample rate and training set size can yield high accuracy