Tatiana Bradley, Rakshith Hiresamudra Shivegowda, and Wai Man Chan

CS 244, Fall 2015

Acknowledgement: Nod Labs, for providing smart ring.
The Problem

**Problem:** Authenticate a user using only sensor data generated by gestures performed while wearing smart ring.

**Why?** Replace (or augment) written signatures by giving immediate feedback on authenticity.

**Diagram of our Proof of Concept:**

- **Nod Ring**
  - Sensor Data
  - Bluetooth LE

- **Client App**
  - Test File
  - Yes or No
  - HTTPS

- **Server**
Goals

**Speed**, to avoid user frustration.

**Security**: signature data not leaked to third party.

**Correctness / Unforgeability**: a real signer has a high probability of being allowed access, while an adversary has a low probability of gaining access.

**No “gesture vocabulary”**: user can use any gesture as a signature, as long as they are able to reproduce it consistently.

**Plane independence**: user can sign on any plane (i.e., a surface or the air), as long as they do multi-plane samples in the training phase.
Hardware/Software Decisions

**Smart ring:** Nod ring from Nod Labs

**Client:** Mac OS Application, adapted from Nod Labs OpenSpatial code

**Server:** Laptop

**Communication:**
Bluetooth Low Energy (Ring ➔ Client App)
HTTPS (Client App ➔ Server)

**Authentication algorithm:** Dynamic Time Warping
We capture accelerometer and gyroscope data from the ring that looks like this:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.059082</td>
<td>0.231934</td>
<td>-0.975586</td>
<td>0.073364</td>
<td>-0.010254</td>
<td>0.303345</td>
</tr>
<tr>
<td>2</td>
<td>-0.048340</td>
<td>0.257812</td>
<td>-1.031738</td>
<td>0.053833</td>
<td>-0.030273</td>
<td>0.282349</td>
</tr>
<tr>
<td>3</td>
<td>-0.062012</td>
<td>0.265137</td>
<td>-1.073242</td>
<td>0.043213</td>
<td>-0.031738</td>
<td>0.273682</td>
</tr>
<tr>
<td>4</td>
<td>-0.095215</td>
<td>0.263672</td>
<td>-1.186035</td>
<td>0.046021</td>
<td>-0.031372</td>
<td>0.286377</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>137</td>
<td>0.502441</td>
<td>-3.038086</td>
<td>-0.579590</td>
<td>-0.283325</td>
<td>-0.188232</td>
<td>0.164673</td>
</tr>
</tbody>
</table>

**Sampling rate:** 100 samples / second

**Challenge:** How to determine how similar one sample is to another?
DTW collapses two time series (of possibly different lengths) into a single distance value that represents the difference between the two time series.

DTW is a dynamic programming algorithm that “warps" two time series on different scales to be on the same scale, and measures the level of distortion required to complete this warping.

It solves the problem that the same gesture could be performed at different speeds in a non-linear fashion.
Our implementation of DTW

Language: C++

Measure of distance: The distance between vectors $(x_1, x_2, ..., x_6)$ and $(y_1, y_2, ..., y_6)$ is:

$$d = \sqrt{\sum_{i=1}^{6} (x_i - y_i)^2}.$$ 

Normalization: Data is normalized to be between 0 and 1 to offset normal variations in size and position.

Asymptotic Running Time: $O(m^2)$ where $m$ is the length of the longer sample.
Client collects user ID and \( n \) samples of same motion from signer and sends to server.

**Server creates array** \( \mathcal{D} \) **of pairwise DTW distances.**

Server **calculates a threshold**

\[
t = \text{average}(\mathcal{D}) + \tau \cdot \text{standard\_deviation}(\mathcal{D}),
\]

where \( \tau \) is a constant tolerance value.

Server associates the threshold \( t \), and the \( n \) reference samples, with the given user ID.

Running time is in \( O(m^2 \cdot n^2) \), where \( m \) is the maximum sample length, and \( n \) is the number of samples.
How we are using DTW: Authentication Phase

Signer gives user ID and does a gesture while wearing ring.

Client collects data and sends the test sample to server.

Server grabs threshold and reference samples for given user ID.

Server runs DTW on the test sample against every reference sample, and finds the minimum distance $d$.

If $d \leq t$, server sends a success signal to the client. Otherwise, server sends a failure signal.

Running time is in $O(m^2 \cdot n)$, where $m$ is the maximum sample length, and $n$ is the number of reference samples.
**Initial Results are Promising**

<table>
<thead>
<tr>
<th>Gesture tested:</th>
<th>Signer’s Initials</th>
</tr>
</thead>
<tbody>
<tr>
<td># of signers tested:</td>
<td>2</td>
</tr>
<tr>
<td># reference samples per signer:</td>
<td>20</td>
</tr>
<tr>
<td># of trials (correct) per signer:</td>
<td>10</td>
</tr>
<tr>
<td># of trials (forgery) per signer*:</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tolerance formula**:</th>
<th>$\text{avg}(\mathcal{D}) + 1.65 \cdot \text{std}_\text{dev}(\mathcal{D})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success rate (correct) :</td>
<td>$18/20 = 90%$</td>
</tr>
<tr>
<td>False positive rate (forgery) :</td>
<td>$1/20 = 5%$</td>
</tr>
</tbody>
</table>

| Est. speed of training:          | 135 ms                                                                   |
| Est. speed of authentication***: | 250 ms                                                                   |

* Forgeries were performed with coaching by real signer.
** Corresponds to 95% confidence level.
*** Includes communication time.
Extensions and Improvements

**Plane independence:** We did some work to allow for user to sign on multiple planes, but this allowed a forger to authenticate on one occasion.

**Testing:** We need to test the system with more users, including users who are not co-creators of the system.

**Security:** We have transport layer security (HTTPS), but there is the possibility that an adversary could gain access to reference samples on the server and trick the system into authenticating.

**Other applications:** The system could be modified to be a non-security-critical gesture recognizer, e.g., to allow a Nod ring user to create custom gestures to do certain tasks.
References

Md Tanvir Islam Aumi and Sven Kratz.
Airauth: Evaluating in-air hand gestures for authentication.

Nguyen Ngoc Diep, Cuong Pham, and Tu Minh Phuong.
Sigver3d: Accelerometer based verification of 3-d signatures on mobile devices.

Lawrence Rabiner and Biing-Hwang Juang.
*Fundamentals of Speech Recognition*.

Software: Nod Labs OpenSpatial, DTW package in R.
Screenshot of App: Successful Authentication

File Upload Successful!
Waiting for Authentication...

Successfully Authenticated at distance = 0.433850
Screenshot of App: Failed Authentication

File Upload Successful!
Waiting for Authentication...

Authentication Failed at distance = 0.863588