CENTRE *for* DOCTORAL TRAINING *in*
CYBER SECURITY

CDT Technical Paper

17/15

Mobile device sensor history as a second factor for authentication

Christian Vaas

Abstract

Despite the availability of biometrics like face recognition and finger print scanners passwords are still a common widely accepted way to secure personal computers and business workstations. This stands in contrast with security measures we can find for cloud services like Dropbox and Google that provide their users with two-factor authentication. Although one could argue that the physical machine loses importance with the shift towards having everything in the cloud ranging from software over platforms to infrastructure there is still a need of securing the endpoint at which these services are accessed. For this purpose, often software or hardware tokens are used along the password in a two-factor authentication scenario. Examples are RSA tokens, software tokens like the Google authenticator or NFC smart cards. A common issue with these solutions is that they are vulnerable to relay attacks. We thus propose a second factor which is resilient to these kind of attacks. To do so it is necessary to guarantee the physical co-presence of the authenticating device and the second factor, e.g. a smart phone. This project aims to ensure this property using the gyroscope history of a smart phone and validating it against received signal strength measurements conducted by WiFi checkpoints within the perimeter, e.g. office building, of the authenticating machine. This makes sure that the smart phone wielder actually walked to the machine and the phone is thus present where the user claims.

1 Introduction

With the raise and the improvement in bandwidth and availability of the Internet more and more of what used to be located on a local machine moves into the Cloud. According to Banerjee et al. [1] 10 years of research indicate that soon everything, ranging from software over platforms to infrastructure, will move into the Internet. Obviously, it is necessary to secure these services from unauthorised access. Many cloud services like Dropbox and Google, therefore, offer the use of two-factor authentication since 2012 [2] and 2011 [3], respectively. Although available, many users do not enable this feature due to usability reasons as found by Gunson et al. [4]. However, even if enabled it is easily circumvented as soon as an adversary gains access to the personal computer of a user. This is because users tend to mark their machines as trusted, reducing the frequency significantly at which they are challenged to enter the second factor. It is therefore crucial to secure the endpoints on which these services are consumed as well.

Locking a computer system with a password, which is still the most commonly used way, cannot be secure enough. Several approaches have been proposed to make these systems more secure. A very widespread one is the use of biometrics like fingerprints or face recognition. Despite the unique character of fingerprints there are still ways of attacking these systems as described by Uludag and Jain [5]. Face recognition systems are not flawless either Galbally et al. show an attack on this biometric in [6]. Another approach, which we promote with this project, is to add a second factor for computer machines to harden these systems. Two-factor authentication, however, is vulnerable to relay attacks which make use of transmitting information over a larger distance as described in [7].

To mitigate the vulnerability to relay attacks we explore the feasibility of sensor data as a second factor of authentication to verify the physical presence of a device. In particular we evaluate the correlation between the history of a smart phone's sensors and the WiFi signal strength measured at WiFi enabled checkpoints, i.e. access points. We then verify the location of a smart phone, which potentially carries a pre-shared secret that is used to unlock a computer. For that purposes we obtain two different sensor time series, gyroscope information and received signal strength.

Moreover, in order to lower the bar for actual implementation of our approach we focus on only using off-the-shelf hardware. Nowadays the sensors provided by a smart phone have a high precision and we believe they are thus accurate enough to be used as a second factor for authentication. The availability of cheap WiFi dongles in combination with boards like the Raspberry Pi enables us to install checkpoints at arbitrary locations. Obviously, these nodes can also be used to set the WiFi infrastructure itself up. This means that instead of installing *normal* access points we would see these smart hotspots which provide infrastructure and server as data acquisition points at the same time.

In our specific approach, the smart phone is in the role of the prover that sends it's recorded gyroscope data to the verifier which is the computer system. We assume that the computer system is in possession of WiFi signal strength datasets as recorded by the access points located within the perimeter. Figure 1 shows an exemplary set-up of the WiFi hotspots and the user's path inside the building. Our hypothesis is that the gyroscope data can be validated using the signal strength from the wireless network as it reflects location information and direction changes as well. It is important to notice that it does not matter which path the user follows it is just necessary that the path described by the gyroscope can be

validated against the WiFi readings. Thus, for the approach to mitigate relay attacks on the computer system. It is necessary to ensure that the authentication token, here a smart phone, is physically present when a user tries to log on to her machine. Obviously, this is only applicable for a scenario where the user performs a local login. We consider remote logins as out of scope for this project as they are also disregarded when considering distance bounding protocols.

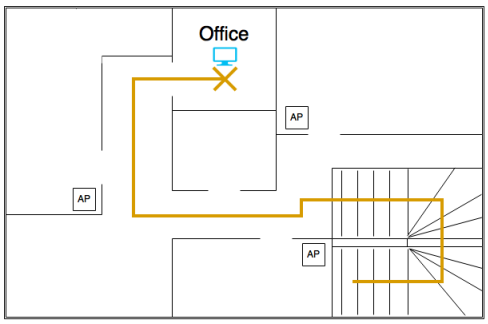

Figure 1: Floor plan - approaching the office

2 Related work

There are already several approaches to address the issue of relay attacks. Many of them try to leverage from the omnipresence of smart phones and the vast amount of sensors available on these platforms. Most of them compute a fingerprint of the environment in order to compare it to either historical data or data recorded with a second device to verify its co-presence.

For instance, Karapanos et al. [8] are using the microphone of a phone and computer in order to record the surroundings and match the two series to verify that they are located in the same place. This approach performs well during their evaluation against Google's two factor authentication mechanism. Further, their user study indicates that people are much more comfortable to use the sound based second factor instead of the authenticator application.

The paper of Shrestha et al. [9] uses an external sensor platform that is connected to the phone in order to expand the different parameters that can be captured by a smart phone. In particular they take readings for temperature, precision gas, humidity and altitude. These series are then sent to the verifier who records the same parameters and sends them to a comparator. A decision about the location of the phone is then made based on the similarity of the readings.

Contextual information of a smart phone are also used by Truong et al. [10] to implement a zero-interaction authentication (ZIA). According to them ZIA usually relies on short range wireless communication that is prone to relay attacks as described in [7] which offsets the usability advantage of these protocols completely. They propose the collection of environmental data to make ZIA more resilient against these kind of attacks. The information they collect is audio, WiFi, Bluetooth and GPS. A verifying terminal then checks these attributes against its own measurements. In case of a match the co-presence of the two devices is assumed.

Shi et al. [11] observe the perimeter of a smart phone to perform continuous authentication of its user. Therefore, they record characteristics for the user like the use of the touch screen, microphone, motion sensor, GPS and the identifier of the cellular mast that the phone is connected to. Their system then continuously monitors these inputs and decides whether a illegitimate user is using it. The observation of fraudulent behaviour logs the out of the system and he is asked for manual authentication.

Whilst all previously mentioned papers make use of a device's sensors there are also other approaches to this problem. Drimer and Murdoch [12], for instance, pursue a distance bounding approach to prevent relay attacks. Their proposal seeks to foil attacks by being able to constrain the round trip time of the communication between a card (prover) and a terminal (verifier). Is this constraint violated the algorithm assumes that a relay attacks is happening and the card is not actually present at the current location. These kind of mitigation techniques rely heavily on timing and bespoke hardware as well as software.

3 Experimental design

The set-up we use for our experimental measurements is displayed in Figure 2. We use two WiFi enabled clients, Node0 and Node1, which connect to the wireless hotspot provided by the smart phone. Both nodes log the received signal strength (RSS) continuously. During the recording of these readings the person holding the smart phone walks on the path as described by the black rectangle facing the direction indicated by the arrows. We thus have four left turns in this set-up labeled numerically from one to four. For the data analysis it is necessary that all participating devices are operating on a shared time line. NTP clients on the respective devices make sure that they operate on synchronised clocks.

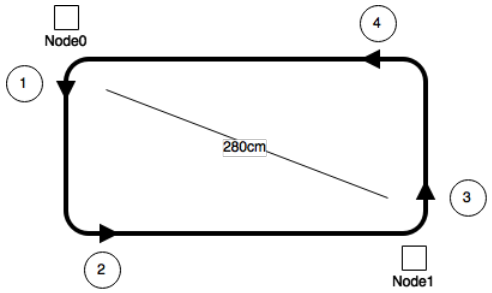

Figure 2: Experimental set-up

3.1 Android application

We chose an Android Nexus 4 as the smart phone because of the ease of development for this platform. Our recording application is written in Java and registers a listener for gyroscope sensor events. These events provide X, Y and Z orientation changes in a coordinate system which is defined relative to the screen of the phone. The x axis is horizontal and points right, the Y axis is vertical and points up and the Z axis points towards the front face of the screen. Each of these events is timestamped upon occurrence and sent to a host via a UDP socket. We use UDP to avoid unnecessary overhead.

3.2 WiFi nodes

On the client side of the WiFi connections we configured a Java application to receive the gyroscope information sent by the smart phone. This information is stored on the machine to be processed alongside the RSS readings.

Separately we collect the received signal strength by running a script that gathers the output of *wavemon* while also timestamping it with the current system time in milliseconds. We can adjust the sampling rate as desired by reducing the delay between these dumps. The script can be found in the appendix.

4 Data analysis

Before we can establish a joint analysis of the RSS and gyroscope data we have a look at each of the time series individually to make sure we operate on the best data possible. First, we acquire a dataset of gyroscope data as displayed in Figure 3. The data seems quite volatile which is due to the phone being placed in the trouser pocket of the carrier. The gait causes the three signals, i.e. the three dimensions, to be distorted.

However, as we can see for the orientation change in the Z direction we can still observe a significant peak. Surely, we could determine the dimension to consider by comparing y value ranges of all three series and then take the one with the biggest difference between upper and lower bound. Another possibility would be to stack all three datasets by taking the maximum at each point in time. Either way we get a series with a lot of noise. For further experiments we hence require the carrier to hold the phone in front of her in a still manner.

For our initial data analysis we test the hypothesis that the human body can improve our results by acting as an obstacle between the WiFi checkpoint and the smart phone. Therefore we perform two experiments. In the first one we undertake two turns, a left and a right turn, with the phone lying on the

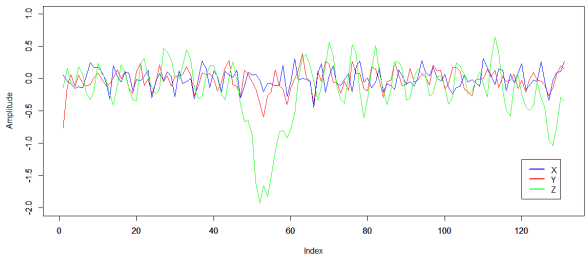

Figure 3: Gyroscope readings while in the pocket

table while we measure the RSS values for the wireless connection. The second one has the modification that the turn is conducted while the phone is held in front of the body. Figure 5 shows the resulting time series.

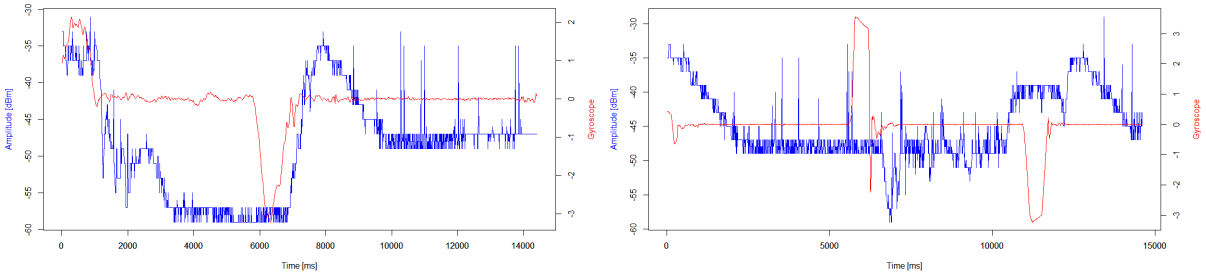

Figure 4: With body (l), without (r)

We now compute the correlation and cross-correlation in order to determine the relation between these signals. Table 1 contains the resulting values with the respective lags for the maximised cross-correlation. As we can see the absolute correlation between the series is much higher when a human body is used to obstruct the WiFi signal. Moreover, we can see that the lag for maximal correlation is 20 which is equal to 171.6 milliseconds. This variation lies within the inaccuracy of our time synchronisation but seems negligible due to the small effect on the correlation values.

	Correlation	Cross-correlation	lag	lag [ms]
Human	-0.230744	-0.235	20	171.6
No Human	0.496347	0.496	0	0

Table 1: Correlation and cross-correlation values

Based on this observation we will have a person carry the phone for further experiments in order to have a more significant change in the signal levels.

In the next step we record the gyroscope and RSS data for the experimental set-up. Figure 2 shows the resulting series. On the right hand of the figure we see the gyroscope reflecting the left turns. Each peak is labeled with the respective number from the design diagram. On the left the RSS readings from of Node0 are displayed. If we now compare the two plots we can see that the peaks of turn number one coincide with the ones in the RSS time series. Like our previous experiment suggested we can see a significant drop in signal quality as soon as a body moves in between the phone and the recording node.

For the further analysis we remove the peaks that are not close to Node0 which results in a time series as depicted in Figure 6.

Moreover, in order to be able to actually compute the correlation between these two datasets we need to first make sure they are of the same cardinality. Since we cannot control the sample rate for the gyroscope, as it is event driven and we can only give the the android system a *hint* [13] for the rate, we have to re-sample the data to make it comparable. We propose a bucket based algorithm that divides the time interval into a given amount of buckets and then tries to assign the data we already have to these buckets. In case there are multiple data points to be assigned to one bucket, we use the mean of these points. Listing 1 shows this algorithm as implemented in R.

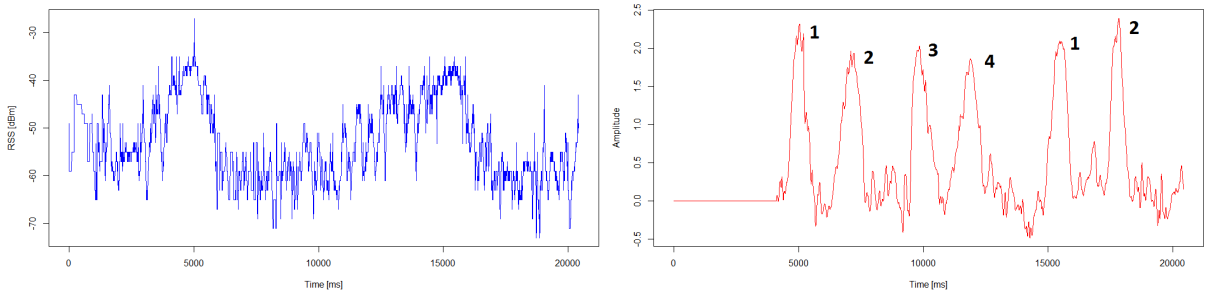

Figure 5: Node0 RSS readings (l), Gyroscope (r)

Listing 1: Bucket algorithm

```
bucket <- function(data, amount, default) {
  duration <- data$Time[length(data$Time)]
  bucketWidth <- ceiling(duration / amount)
  timeRes <- rep(0, amount); valueRes <- rep(0, amount)
  for (i in seq(1, amount)) {
    bucketTime <- i * duration / amount
    timeRes[i] <- bucketTime
    lb <- bucketTime - (bucketWidth / 2)
    ub <- bucketTime + (bucketWidth / 2)
    valueRes[i] <- mean(data$Value[data$Time >= lb
                        & data$Time < ub])
    if (is.na(valueRes[i])) {
      valueRes[i] <- default
    }
  }
  return(data.frame(Time = timeRes, Value = valueRes))
}
```

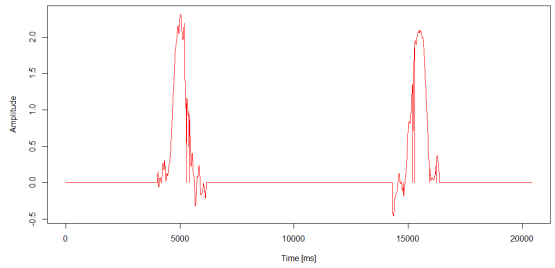

Figure 6: Turns close to Node0

Choosing an amount of buckets that is too big so that the algorithm cannot assign a value to each bucket leaves holes in our dataset. We explore two different approaches to solve this problem.

The first technique uses four different values to generate a value which is then provided as the *default* argument of the algorithm. The values we consider are:

- Most common value (MCV)
- Mean over all readings
- Median over all readings
- A constant, e.g. 0

After having adjusted the two datasets we can compute the correlation for each combination of the above four value generation strategies. Table 2 displays the different correlation values. Rows indicate

the configuration used for the RSS data while columns represent the gyroscope readings. We can see that using the mean for both series gives us the highest correlation.

	MCV	Mean	Median	Zero
MCV	0.5145898	0.5149698	0.5141426	0.5141426
Mean	0.5152189	0.5155974	0.5147731	0.5147731
Median	0.5150745	0.5154535	0.516282	0.5146282
Zero	0.5004783	0.5008282	0.5000616	0.5000616

Table 2: Correlation results

The second approach we evaluated is to use an algorithm that automatically reduces the amount of buckets until we do not have any gaps in the re-sampled time series. This preserves as much information of the original series as possible while still achieving our goal of having identical sampling rates for all our datasets. Moreover, we do not have to add interpolated data. We compute the maximum bucket amount for both measurements and then use the minimum of this to create two series with equal length. In our specific case this leaves us with 688 buckets, i.e. 688 data points, which is a reduction by 2067 and 3311 for the gyroscope and signal strength readings, respectively. The calculated correlation for this technique yields a slightly higher value as displayed in Table 3 between the two series.

Of course, this correlation value on it's own does not give any foundation for deciding whether we could successfully verify the movement information against the received signal strength. Hence we need to use a different pair of left turns that have been recorded close to Node1 which should barely coincide with the drops in the RSS readings from Node0. The resulting correlation value for the two series we expect not to be related is significantly lower than the one calculated for the matching measurements. See Table 3.

Default value	Adjusting amount	Not correlated	Stacked RSS
0.5155974	0.531269	0.1490527	0.4553815

Table 3: Algorithm performance

Further, under the assumption that we might be able to observe this behaviour for all turns visible in the gyroscope information we use the signals recorded from the four corners of the path outlined in Figure 2. These four series are combined to one by taking the maximum for each point in time. The result is plotted in Figure 7. We use the max operator here after having tried to use the summation of the signal which caused the result not to be meaningful because of the signals cancelling each other out. If we now compute the correlation between the initial gyroscope data and the combined series we get a value of 0.4553815, as shown Table 3, which is a bit lower than the one for the turns in Figure 6 but still distinct from the one obtained for the non matching curves.

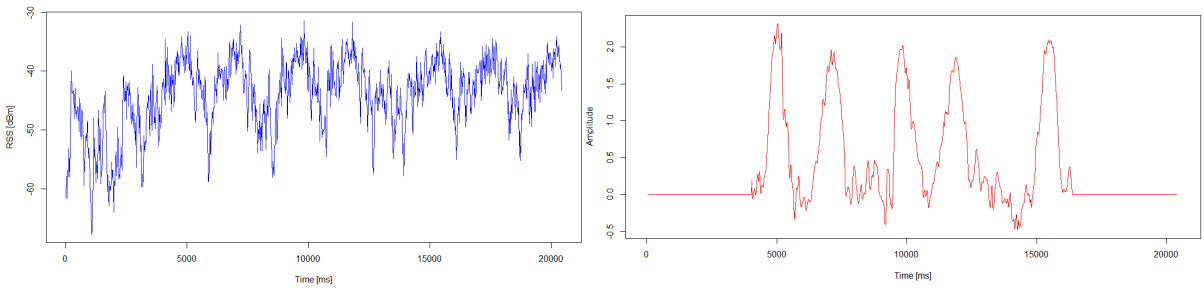

Figure 7: Stacked RSS series (l), Gyroscope data of five turns (r)

In order to see how this analysis performs on non matching curves we now cross-combine our datasets. First, we take the gyroscope data from Figure 6 (Dataset A) and compute the correlation with the stacked RSS series (Dataset B), see Figure 7 left. Next, we compute the correlation between the five turns as displayed in Figure 7 (Dataset C) and the received signals strength from Figure 5 (Dataset D). The results are then compared in Table 4. As we can see the values for the time series that should match, coloured in blue, are almost double the correlation for the ones that should not match.

	Dataset A	Dataset C
Dataset B	0.2789674	0.4553815
Dataset D	0.531269	0.1972083

Table 4: False dataset correlations

5 Future work

In order to do a full evaluation of the approach we propose further research for a deeper understanding of the topic and the issues that come with it. We see huge potential by investigating different walking patterns as well as performing a user study. Within this study an assessment of the usability of such a system should be conducted. Since the enrolment process does not require the user to perform extensive tasks we would expect the usability of the system to be rated quite good. However, it is also necessary to see how comfortable users would feel to use the system. This can be affected by privacy concern or personal attitude towards technology in general. Also, the aspect of having a backup solution in case the user's phone runs out of battery or gets stolen might influence the experience with the system. In common two-factor authentication systems this is solved by providing the users with a set of emergency codes they can print out and store in a secure place.

Regarding the data analysis part of this project we believe an assessment of the false acceptance rate (FAR) as well as the false rejection rate (FRR) should be done for several scenarios. The false acceptance rate is the frequency by which the system accepts authentication attempts that it should not whereas a false rejection occurs when the system does not grant a user access although it should. Afterwards, the equal error rate (ERR), which is the point where the FAR is equal to the FRR, can be used to determine an adequate threshold for the correlation value. Instances that should be considered are a single scenario with multiple test subjects, i.e. different people, and different scenarios which are then used to cross match data sequences.

In general, it is worth expanding more on different techniques to measure the relation between the RSS data and the gyroscope readings. We could, for instance, think about translating the series to character sequences that are then compared using diversity measures like the Hamming distance [14]. Based on that we could again compute the FAR and FRR to determine the feasibility for authentication.

Further ,we have not taken the floor plan into account so far. Obviously, the gyroscope would reflect a possible path for the layout of the building obeying constraints like walls and rooms. However, checking the gyroscope data against the floor plan alone would surely not be good enough as this information is easy to imitate given that some buildings are public or people could pay cleaners to get knowledge about the layout of a building. But it is additional information that could be used to be incorporated and thus strengthen our approach.

Taking the floor plan into account also gives us distance information for the different segments a person has to pass before getting into her office. We could compare this to the walking distance captured by the phone. As described by Sagawa et al. [15] it is possible to measure this distance using a three dimensional accelerometer. This type of sensor is available through the Android API [16] and could thus be easily accessed and used for our purposes. However, further research into this needs to be conducted before we can estimate how beneficial this proves for our approach.

It might also be worth considering how much information we can infer from the WiFi connection established with the checkpoints inside the building. For instance the SSID of the WiFi or the password of the WiFi could be used for additional security measures. Moreover, we need to think about whether it poses a thread if the WiFi is broadcasting its SSID or not. In general this feeds into the last point for this section.

It is absolutely critical to perform a proper security analysis of the authentication method. This includes defining a reasonable thread model and its evaluation as well as the analysis of the used hardware and software technologies. Especially protocols like NTP which is necessary to guarantee the time synchronisation of the data we use needs to be investigated further. According to Bishop [17] there are security flaws that need to be addressed. In case there is no solution for the time synchronisation we have to think about other possibilities and might need to resort to dynamic time warping (DTW) for indexing time series using an algorithm as described by Keogh and Ratanamahatana [18]. This on the contrary might give adversaries an additional attack surface.

6 Conclusion

We have established a basis on which we can perform experiments to gather received signal strength as well as gyroscope readings from a mobile device. Further, we laid the foundations to do data analysis on the collected data and understood the structure of this data. The results we got so far seem promising and we could find a correlation between the RSS measurements and the orientation changes of a smart phone, which are reflected by gyroscope changes. However, a pure correlation on the positive matches, i.e. the cases in which we want to successfully authenticate a user, cannot be used for a real world system. We thus also briefly investigated cases in which we probe gyroscope data that should not be verifiable by the used RSS data. At this, we could observe significantly lower correlation values and are hence confident we will - with further research and development - eventually propose a system that covers our use as a second factor in authentication to a satisfying degree.

Acknowledgement

This project was supervised by Dr. Ivan Martinovic. I want to thank him for his time and the various valuable meetings. The EPSRC funded this research as part of the CDT in cyber security.

References

[1] P. Banerjee, R. Friedrich, C. Bash, P. Goldsack, B. Huberman, J. Manley, C. Patel, P. Ranganathan, and A. Veitch, "Everything as a service: Powering the new information economy," *Computer*, no. 3, pp. 36–43, 2011.

[2] I. Dropbox, "Another layer of security for your dropbox account," August 2012.

[3] I. Google, "Advanced sign-in security for your google account," February 2011.

[4] N. Gunson, D. Marshall, H. Morton, and M. Jack, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Computers & Security*, vol. 30, no. 4, pp. 208–220, 2011.

[5] U. Uludag and A. K. Jain, "Attacks on biometric systems: a case study in fingerprints," in *Electronic Imaging 2004*, pp. 622–633, International Society for Optics and Photonics, 2004.

[6] J. Galbally, J. Fierrez, J. Ortega-Garcia, C. McCool, and S. Marcel, "Hill-climbing attack to an eigenface-based face verification system," in *Biometrics, Identity and Security (BIdS), 2009 International Conference on*, pp. 1–6, IEEE, 2009.

[7] M. Roland, J. Langer, and J. Scharinger, "Practical attack scenarios on secure element-enabled mobile devices," in *Near Field Communication (NFC), 2012 4th International Workshop on*, pp. 19–24, IEEE, 2012.

[8] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Sound-proof: Usable two-factor authentication based on ambient sound," *arXiv preprint arXiv:1503.03790*, 2015.

[9] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Drone to the rescue: Relay-resilient authentication using ambient multi-sensing," in *Financial Cryptography and Data Security*, pp. 349–364, Springer, 2014.

[10] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, "Using contextual co-presence to strengthen zero-interaction authentication: Design, integration and usability," *Pervasive and Mobile Computing*, vol. 16, pp. 187–204, 2015.

[11] W. Shi, F. Yang, Y. Jiang, F. Yang, and Y. Xiong, "Senguard: Passive user identification on smartphones using multiple sensors," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on*, pp. 141–148, IEEE, 2011.

[12] S. Drimer, S. J. Murdoch, *et al.*, "Keep your enemies close: Distance bounding against smartcard relay attacks.," in *USENIX Security*, vol. 2007, 2007.

[13] G. Inc., "Sensormanager," June 2015.

[14] Wikipedia, "Hamming distance," June 2015.

[15] K. Sagawa, H. Inooka, and Y. Satoh, "Non-restricted measurement of walking distance," in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, vol. 3, pp. 1847–1852, IEEE, 2000.

[16] G. Inc., "Sensors overview," June 2015.

[17] M. Bishop, "A security analysis of the ntp protocol version 2," in *Computer Security Applications Conference, 1990., Proceedings of the Sixth Annual*, pp. 20–29, IEEE, 1990.

[18] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

Appendix

Listing 2: RSS measurements script

```
#!/bin/bash
timestamp=$(date +%s)
folder=logs_$2_$timestamp
mkdir $folder
for i in `seq 1 $1`;
do
  echo $i
  time=$(date +%s%3N)
  wavemon -d >> $folder/log_$i.txt
  echo "Time: $time" >> $folder/log_$i.txt
  sleep 0.005
done
cat $folder/log* | grep "signal\|Time" > $folder/$2.txt
rm $folder/log*
```