Gait-Based Identification Using Wearables in the Personal Fog

Charles Walter
University of Tulsa
charlie-walter@utulsa.edu

Rose F. Gamble
University of Tulsa
gamble@utulsa.edu

Abstract

Wearables are becoming more computationally powerful, with increased sensing and control capabilities, creating a need for accurate user authentication. Greater control and power allow wearables to become part of a personal fog system, but introduces new attack vectors. An attacker that steals a wearable can gain access to stored personal data on the wearable. However, the new computational power can also be employed to safeguard use through more secure authentication. The wearables themselves can now perform authentication. In this paper, we use gait identification for increased authentication when potentially harmful commands are requested. We show how the relying on the processing and storage inherent in the personal fog allows distributed storage of information about the gait of the wearer and the ability to fully process this data for user authentication locally at the edge. While gait-based authentication has been examined before, we show an additional, low-power method of verification for wearables.

1. Introduction

Wearables are becoming ubiquitous for consumers. Smartwatches, wireless headphones, fitness trackers, and even medical wearables, such as insulin pumps and heart rate monitors, are becoming commonplace in the lives of millions of consumers. These devices collect significant data about the user. Heart rate, movement, location data, activity level, and, in the case of medical wearables, private medical data about the user. The amount of data collected makes wearables an enticing target for attackers.

While the ubiquity has increased, so too has the power and storage of the wearables themselves. Devices like the Apple Watch 3 [1] or Samsung Gear [19] are both capable of performing major processing for apps that can be loaded directly into the internal storage of the watch. We also see increased power in devices like the Here One [7] that can process audio and remove or amplify specific sounds from a user's environment. This shows that wearables, as is the case with most technology, are moving towards the point where wearables are as capable as our current phones. In fact, the Apple Watch has about the same processing power as the iPhone 4 [9].

With the increase in processing power of the wearables, it becomes possible to make the wearables be edge nodes in a fog architecture. When combined with an additional base station layer, this architecture becomes a personal fog [22], in which all fog nodes are owned by the user. By using the personal fog, it is possible to process the data collected by the wearable's sensors directly on the wearable and make additional decisions, either for security or for app functionality. The computational load can also be shared by all peer fog nodes, rather than only the wearable or only its base station.

Unfortunately, with the increased power, data collection, and other functional capabilities of wearables, there is a risk that, if the wearable falls into the wrong hands, it could allow an attacker to gain personal information about the original user. For example, Android Smart Lock [10] allows devices that have been declared by the user to be trusted to unlock the users' phone. Thus, as long as the trusted wearable is within Bluetooth range the phone will be unlocked. This accessibility poses a serious security risk. An attacker needs only to steal a trusted wearable and the phone to gain access to all data stored on the phone.

The idea behind the Android Smart Lock system is to allow the user to have increased privacy without compromising convenience. However, especially with the trusted devices option, this feature goes too far in opening the door for attackers. It would be better for the user if there was a method of using their data from their wearables to authenticate the user without the user needing to perform any major action, but in a way that, if an attacker managed to gain access to the wearable and base station, they would not be able to access the user's personal data.

In this paper, we use a user's gait for authentication when attempting to perform tasks that may be harmful to the user should an attacker have access to the wearable/base station. We choose gait as our authentication method because it is unobtrusive for the

URI: https://hdl.handle.net/10125/60159
ISBN: 978-0-9981331-2-6

user (a user does not need to perform any additional identification beyond standard use), is easily collected with existing wearables and base stations, and has been used by other researchers as a method of user identification with high accuracy. We propose the use of the personal fog to distributed stored data on each fog node to all fog nodes in the system, allowing independent verification by each fog node to ensure a potentially compromised fog node does not allow an attacker to gain access to a user's personal data. We show that Pearson correlation can be used as a low-computational cost method of authentication and confirm that the additional time to verify is negligible for the user.

2. Background

Gait-based authentication has been examined by other researchers. Boyle et al. [3] used the Euclidean norm of accelerometer readings and a k Nearest Neighbors algorithm to identify users' gaits. They were fairly accurate, though in some cases they had an accuracy of as low as 70%. Sharma et al. [20] used image processing techniques to identify walkers with a 97.5% accuracy. Papavasileiou et al. [18] used "Smart Socks" to authenticate users by their gait. They achieved perfect recognition, though they were only comparing between the two socks. Ho et al. [8] used a phones accelerometer to detect user's gait. They combined the data from the x, y, and z axes and used a Bayesian classification to identify the user. In most cases, they achieved an accuracy of between 69.7 and 100%.

Xu et al. [23] used a smartwatch for gait authentication. Their method requires significant computation to use, as it performs pre-processing and focuses on identifying walking, running, and idling for its identification. Muaaz and Mayrhofer [14] used adapted Gaussian mixture models to identify users based on their gait from a cell phone. Their method also requires significant computation, as they omit unusual walking cycles and estimate the user's gait from the actual data. Their method does allow for orientation independent verification, however [15]. Cola et al. [5] used a wrist-worn device, simulating a smartwatch, for their gait analysis. Their method depends on preprocessing, feature extraction, and anomaly detection, producing an accuracy between 97.3 and 99.6%.

Gait has been used for purposes other than identification. Hwang et al. [11] examined gait to measure walking quality using an Arduino attached to the user's leg. They were able to fairly accurately (between 81.6 and 95.8%) identify different walking styles. Xu et al. [24] used gait to generate secret keys between wearable devices and their base stations. Their method identified the heel strike, which can be identified by all wearables on the body, to generate secret keys for encryption of Bluetooth traffic. They were able to generate keys with an accuracy between 72.1 and 98.3%. However, eavesdroppers were able to generate keys accurately approximately 50% of the time, reducing the usefulness of the method.

The fog, despite it being a relatively recent computing architecture, has been used for a large number of applications. These applications include providing resilience at scale [4], robotics ([6], [12]), data analysis [21], and social sensing for limited internet connectivity [16]. It is likely the fog could target many applications that require additional computing resources, making it ideal for mobile and Internet of Things applications.

3. The Personal Fog

In this section, we describe how wearables can interact within the personal fog. In the personal fog described in [22], wearable devices have additional processing and storage capabilities. This assumption reflects the increasing capabilities of devices such as the Apple Watch and the Samsung Gear, which both contain additional storage space and processing power for data collection and housing developer apps. Each wearable is treated as a fog node at the edge, taking information from the built-in sensors, performing basic processing and compression, and forwarding that data to their base station, in this case a phone. While not all wearables contain this additional processing power, the personal fog invokes a trend of increasing processing power to the edge in recent years and makes the assumption that all wearables will be powerful enough to function as a fog node.

With the expected increased computational power of the wearables, it becomes possible for wearables to reason about their environment without relying on their base station. In a traditional wearable architecture, the base station is in complete control of the wearable and, if the wearable is capable of controlling the base station (as is the case with smart watches being able to change/pause music or unlock the phone), the base station must relinquish its control to the wearable. There is no additional verification performed to ensure the wearable is not being used by an attacker to control the phone. With the personal fog, and the additional computational power on the edge, the wearables can make local decisions, based on their sensor data, to verify the user.

The architecture of the personal fog is depicted in Figure 1. Note that the wearables act as a fog node connected directly to their sensors. The wearables are connected directly to the base station, which can communicate directly with the cloud. By making the wearables their own fog nodes, it becomes possible to maintain the structure of a fog while ensuring that all nodes except the cloud are owned by the user, creating a true personal fog.

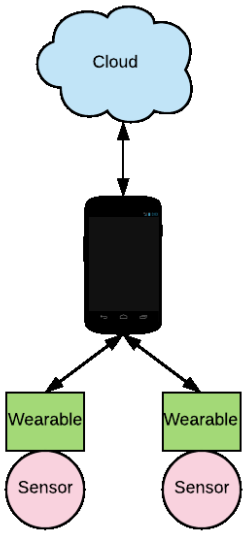

Figure 1. The Personal Fog Architecture

We place a common app on each personal fog node, i.e. the wearables and the base station. This app can process the collected data at the edge wearable in the personal fog to determine if the user is in an insecure environment, as described in [22]. The base station can also perform the data collection and determination. The wearable analyzes only its own data and compares it against internal rules that assess the environmental parameters and determine the security status. The base station aggregates data from all connected wearables to perform its own determination of the current security status. If either denote insecure as the status, then the devices react to that state until a secure status is verified.

Imagine a user has a Garmin Smartwatch. This watch allows the user to set up the watch to reply to text messages, unlock the users phone, and view notifications, even if the phone does not show the full notification on screen. If an attacker gets access to the watch and the base station, the attacker will gain access to the user's phone, even if the user has a strong password or fingerprint verification enabled. If the user did not set up the watch to be able to unlock their phone, the attacker will still be able to read messages and reply to texts as the user. Such scenarios can be prevented by the additional power granted to wearables by the personal fog, allowing the wearable to locally process and verify the user based on data collected by its own sensors or peer wearables.

The architecture of the personal fog is not required for the wearable to verify with only its own data. However, this verification method could be faked by an attacker. The attacker needs only to get root access to the wearable and force the stored data used for verification to be replayed. With the interconnectivity of the personal fog, the verification data can be spread across all devices, where each device can verify all other device data. In this way, authentication is performed by all devices and, should the user fail authentication, the fog nodes can shift into an insecure state to prevent further attacks.

It should be noted that, for the purposes of this paper, we focus only on the interconnectivity between the wearables and the base station for authentication. While the cloud is capable of informing the base station of its state and collecting data from the base station, there are existing methods of verification for cloud communication [2]. We do not propose the use of gait information when communicating with the cloud, though this may be implemented in the future. For the purposes of this work, the cloud only aggregates information about the current security state of the user and is responsible for informing the user when they enter an insecure state.

4. Collecting Valid Gait Information

For authentication, we chose to use accelerometer data of a user walking to verify using the wearer's gait. Wearables already collect this data, often using the accelerometer data to control functions of the device (smartwatches that light up when the wrist is flicked, the Apple Watch that opens Siri when watch is brought up to the users' mouth). With the additional edge power granted as part the personal fog architecture, we can perform additional processing and storage of this data collected locally by the wearable. The wearable, running our personal fog app, can identify when the user is walking, collect valid gait information for storage and later use, and adapt to potential attacks if authentication fails.

We focus on a user's gait primarily because it is unobtrusive to collect for a user of the personal fog and its accuracy in identification. Other biometric options are available, including facial recognition, retinal patterns, fingerprints, speech recognition, or facial thermograms [13], but each requires additional work by the user and additional hardware to be implemented by device manufacturers. For facial recognition or

retinal patterns, cameras must be installed in all wearables and base stations and the user must raise each device to their face for identification. A similar issue exists for fingerprinting and facial thermograms. Speech pattern recognition could work, though any attacker who has a recording of the users' voice could perform verification and user verification may not be possible in noisy environments.

There are wearables, such as the Nymi band [17], that use ECG sensors to verify a user by their heartbeat. This sensor could possibly be used as a more secure authentication, but a user is required to press their finger into the device to perform the ECG reading, adding an additional step, and only the single device can be used. Until more wearables make use of an ECG, as well as find a way to eliminate the additional user step, it is infeasible for the average consumer to use the device to ensure their information is secure. Thus, for use with the personal fog, gait is a very feasible option requiring no additional steps for verification (save for actual steps when walking) and no additional burden on device manufacturers.

In order to authenticate a user by their gait there must be consistent "valid" data collected. This collection should be performed by the user when they first set up a new device. It is reasonable to assume that when a user adds a fog node to their system they are a valid user.

When a user first sets up their device, our app looks for accelerometer data that is consistent with walking. We specifically look for long stretches of rhythmic jumps in the accelerometer data that imply footsteps. This examination is done by searching for peaks and valleys in the data that are relatively close to each other (local maximums and minimums within a small error window of about 0.05). We define a valid amount of data to be greater than 5000 data points, equivalent to 20 seconds of walking.

We then trim this data to the middle 1100 data points. This cleaning of the data ensures that the data we rely on is not from the very beginning or very end of a walk, as in practical tests this data was inconsistent with normal walking. This reason is because users being studied would walk slightly more irregularly at the very beginning or end of their trip, perhaps as a result of knowing they were carrying a wearable for testing. It may also be due to a user needing to take a number of steps to reach a natural stride. We chose to store 1100 data points so that we can choose 100 data points for end verification with a starting point anywhere between the start and the 1000^{th} data point, allowing some variation in the start and end points during verification. This choice is not needed but adds an extra layer of complexity that wearables can use to verify. When verifying, fog nodes can choose any point within the 1100 datapoints to use for verification, provided it meets the requirements for verification, which are discussed in Section 6.

Once consistent data is collected, it is temporarily stored. This stored data is then verified as being accurate the next time a walk is detected. If verification succeeds, the data is distributed to the other fog nodes for storage and to be used by the other fog nodes when verification is required. Once a confirmed consistent walking pattern is stored on other devices, fog nodes store only the most recently consistent walking pattern. This peer storage method allows a wearable to send the most recently collected gait data to be used when authentication is required and a user is not walking. It is likely to be the case often, as users tend to be stationary for longer periods of time than they are walking.

5. Distribution of Gait Data

Both wearables and phones, acting as the wearables' base station, are constantly collecting accelerometer information, and may tell the user to move if the user has been stationary for too long. Many wearables use their accelerometer data to calculate steps or to recognize the orientation of the device. With this data, it becomes possible to create a "walking profile" of the user.

Because wearables are worn on the same parts of the body (a smartwatch is usually worn on the same wrist at all times) and the base station is often stored in a consistent location (pocket, external bag), gait information from each device will be consistent for each user. Thus, there is a reasonable expectation that a user will provide consistent gait information across their devices when they are walking. While specific devices may differ (the base station will have different accelerometer readings than a smartwatch, for example) the data collected from a single device will be consistent with that device.

When a wearable attempts to perform an action that the system or the user has determined to be a potential security risk, such as unlocking the users phone or sending a reply message, the wearable attempts to authenticate with the other nodes in the users personal fog. Figure 2 shows the flow of this authentication. First, the wearable requests verification from all connected nodes in its personal fog. Once the nodes have responded that they perform the requested verification, the wearable sends its most recently collected valid gait data. The fog nodes then verify the gait data they receive from the wearable. If the gait data is valid, a fog node sends back a "True" value to the wearable.

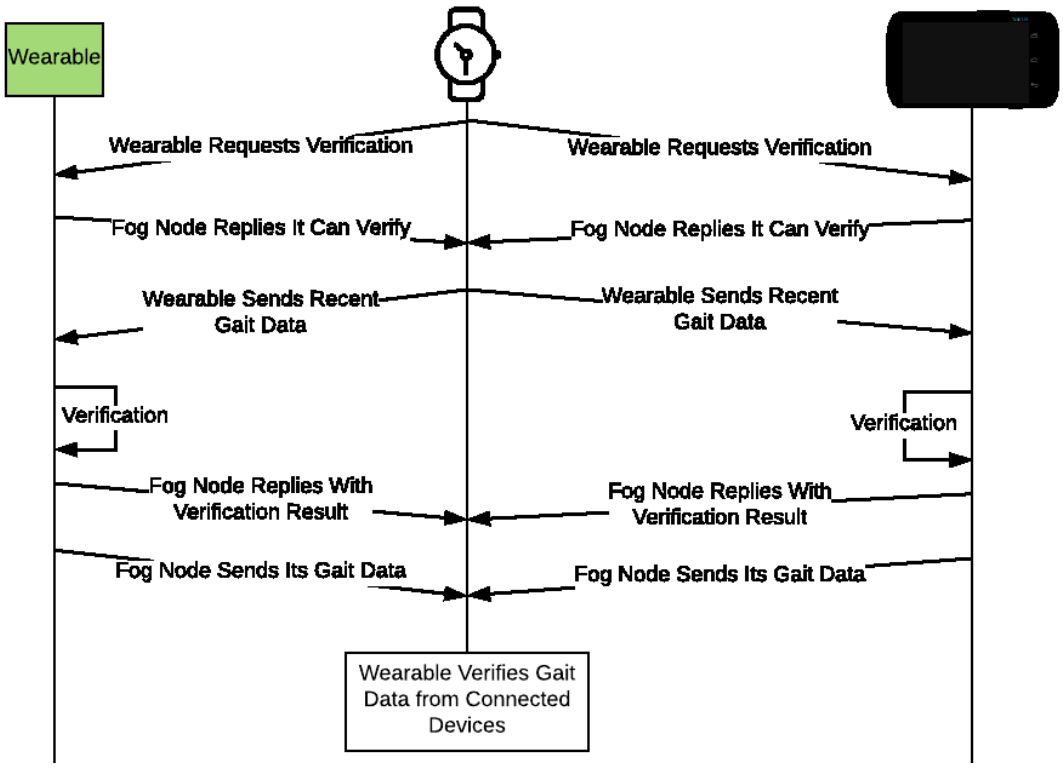

Figure 2. Communication Flow for Gait Verification

The wearable cannot just accept this true value as being valid, however. An attacker may attempt to fool the device by intercepting the transmissions using a man-in-the-middle attack and send back a "True" value regardless of the actual verification result. To prevent this spoofing, the other fog nodes also send their most recent gait data to the wearable. The wearable then uses its stored data from those devices to validate each device. Once validated, the wearable can assume the previous response is correct and, if the user has been authenticated, perform the action the user attempts. If the response is not validated or the wearable receives a "False" value, it blocks the action and shifts into an insecure state, preventing additional attacks.

There can be an issue when the wearable is connected to a new base station and does not yet have a copy of the base station's gait profile. In this case, if there are no other peer wearables available in the personal fog, the wearable will assume it is not allowed to perform any of the potentially harmful actions that require authentication. While this problem means the user would be unable to use their device without additional authentication methods, it is temporary, since the base station will provide the wearable with gait data once the user has walked with the new base station.

6. Authenticating Gait Data

To authenticate the gait data, we propose a simple method based on Pearson Correlation. By correlating the data in this way, we reduce the computational power needed to perform the verification so that it can be performed by a wearable. Other methods, such as those described in the background, can be used as well.

For our method, we take the first 1000 datapoints and find a series of datapoints from the middle of a walking cycle. We select the next 100 datapoints for validation. More datapoints could be used, but a smaller number allows for more variation in speed of walk cycles, as a user may have slowed down or sped up within a walk. Minor variation could cause issues with the correlation for a larger number of datapoints. We then use the 100 datapoints directly following the initial peak. This separation allows our method to always begin on a peak and makes it significantly more likely to begin on a consistent walking cycle. We then run a Pearson correlation on the 100 datapoints.

One problem with running a pure Pearson correlation on gait data is that attackers could correlate with the user by virtue of walking "together". In such a case, a

Pearson correlation will show a significant correlation ($p < 0.01$) between the two users because both users will have similar, though slightly different, peaks and valleys just from the act of walking to maintain the same stride. This situation is obviously a problem with using Pearson correlation for verification.

To prevent this issue, we look only for data which has an r-value above a preset threshold. The r-value is used is a value between -1.0 and 1.0 that represents how associated two variables are. A value of -1.0 shows there is a negative relation, while a value of 1.0 shows they are completely related. A value of 0.0 represents no relation. Correlation between two different gaits is inevitable, but the degree of correlation is not. By specifying a threshold that must be met to authenticate a user, we can ignore even strong correlations that may arise as a result of both user's gaits being correlated based only on both sets of walking data. We choose a threshold of 0.70 for our tests, though modification by the user is an option for increased security.

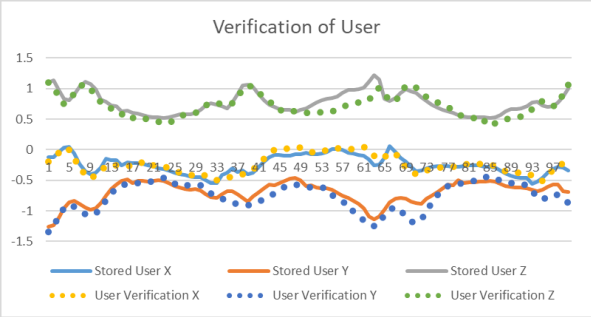

Figure 3. Verification of Gait by Legitimate User

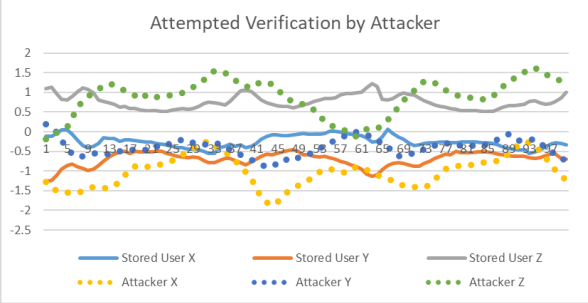

Figure 4. Attempted Verification of Gait by Attacker

7. Evaluation

To evaluate our system, we conducted a small test with 6 attackers and 5 legitimate attempts at verification. For each test, users, acting as an attacker, were asked to hold a Raspberry Pi, acting as a wearable, in their right hand to mimic a smartwatch. They then walked approximately 3000 feet to measure their gait. All attackers walked the same route that the verification data was collected on. Attackers ranged in height and gait-length, with one attacker having the same height and gait-length of the legitimate user and the other attackers having a smaller height and gait-length. For the verification of the original user, our legitimate user walked the initial route to collect the verification data. The user then walked the same route on a different day and 4 different routes at different times over the course of a week.

A graphical representation of one legitimate verification attempt can be seen in Figure 3 and one attacker attempting to verify can be seen in Figure 4. The stored gait data is shown in solid lines, while the data being verified is shown in dashed lines.

These graphs are intended to show that, at a glance, the gaits are different enough to say they are indeed different users. To show that verification worked as intended, we then ran a Pearson correlation on each of the attackers and legitimate user data. Results of verifying the legitimate user can be seen in Table 1, while attempted verification of the attackers can be seen in Table 2.

When examining these tables, the methodology looked for verification between X values, Y values, and Z values, based on the assumption that the wearable is held in the same orientation in the same hand across tests. This assumption reflects common usage since users wear their devices in approximately the same area. Thus, when looking for correlations, we are only looking at the values starting from an X correlation and going diagonally down to the Z correlation. For example, the X values of each walk are correlated with the stored X values, the Y values are correlated with the stored Y values, and the Z values are correlated with the stored Z values. Correlating X values with Y or Z values are not examined.

In Table 1, we can see that all additional walks correlate strongly with the stored gait data. The lowest correlation, Walk5 Z, still correlates at a value of $r = 0.71$ with Stored Z. This gives a p-value of $p < 0.00001$. Of note, however, is that any r value greater than 0.256 will give us a p-value less than 0.01. We expect this result, as any two users walking will have a correlation that they are walking. Our methodology relies on incredibly high r values to identify a user by their gait. Interestingly, the Y and Z values are all highly negatively correlated. For example, Walk5 Z is negatively correlated with Stored Y at $r = -0.71$. This result could be added into the verification process in the future as an additional check to

authenticate a user, though more research would be needed to confirm it is the case for all users. It is possible that it is just a result of the style of gait that the legitimate user has.

Table 2 indicates that none of the attacker's gaits correlate at a level we expect for verification. The highest correlation comes from attacker 4, with both of their X and Y values correlating at a value $r > 0.5$, and from attacker 6, who had a Y value correlated at $r = 0.65$. These values are lower than the 0.71 minimum found with a legitimate user and below the 0.7 valued expected for verification. Interestingly, the highest correlations came not from correlating the X, Y, and Z values with the users corresponding X, Y, and Z values, but with correlations between X, Y, and Z values. For example, Attacker 5's Z value had a strong negative correlation (-0.75) with the user's X value.

It is possible that an attacker can try to mimic the user's gait, especially if the attacker is familiar with their target. To examine this scenario, we recruited two subjects, one of the same height and leg length as the legitimate user and one of a different height and leg length, attempting to mimic the user's gait. Both walked directly next to the original user, allowing them to ensure their steps matched the user as closely as possible. Each attacker attempted to mimic the user's gait twice.

Table 1. Pearson Correlation of Legitimate User

	Stored X	*Stored Y*	*Stored Z*
Stored X	1		
Stored Y	-0.13462	1	
Stored Z	0.20493	-0.85757	1
Walk1 X	0.921468	-0.06334	0.161636
Walk1 Y	-0.21378	0.919074	-0.90257
Walk1 Z	0.056176	-0.76916	0.871817
Walk2 X	0.899738	-0.19281	0.259065
Walk2 Y	-0.12032	0.817692	-0.8325
Walk2 Z	0.009094	-0.80609	0.897764
Walk3 X	0.915283	-0.27262	0.330213
Walk3 Y	-0.20816	0.907648	-0.87047
Walk3 Z	0.132923	-0.85172	0.921374
Walk4 X	0.812616	0.039542	0.160344
Walk4 Y	-0.24806	0.873684	-0.85932
Walk4 Z	0.084646	-0.7836	0.83035
Walk5 X	0.890756	-0.29117	0.418455
Walk5 Y	-0.3287	0.845019	-0.71493
Walk5 Z	0.200942	-0.71473	0.717343

Table 2. Pearson Correlation of Attacker

	Stored X	*Stored Y*	*Stored Z*
Stored X	1		
Stored Y	-0.13462	1	
Stored Z	0.20493	-0.85757	1
Attacker1 X	-0.6623	0.325599	-0.54795
Attacker1 Y	-0.14655	0.474359	-0.33881
Attacker1 Z	0.462857	-0.52557	0.491506
Attacker2 X	0.048722	-0.27774	0.071827
Attacker2 Y	0.314086	-0.42231	0.221583
Attacker2 Z	-0.36024	0.305483	-0.34226
Attacker3 X	-0.12285	-0.58759	0.545348
Attacker3 Y	0.196888	-0.51463	0.415218
Attacker3 Z	0.166804	0.4271	-0.49472
Attacker4 X	0.569107	-0.18589	0.220556
Attacker4 Y	0.155689	0.552398	-0.32765
Attacker4 Z	-0.13	-0.45604	0.240945
Attacker5 X	-0.65764	0.246436	-0.35979
Attacker5 Y	-0.10304	-0.3374	0.055477
Attacker5 Z	-0.7479	0.502703	-0.43321
Attacker6 X	-0.24377	-0.54304	0.430248
Attacker6 Y	-0.55745	0.649798	-0.71654
Attacker6 Z	-0.17141	0.30546	-0.43277

Table 3 shows the result of the attackers attempting to mimic the user's gait. Mimic 1 and 2 are the attacker of the same height and leg length and Mimic 3 and 4 are the attacker of a different height and leg length. Interestingly, the highest correlation at $r = 0.57$ occurred with the attacker of a different height and leg length. If we allow the Z direction to be correlated with Y, the attacker of the same height and leg length has a maximum correlation of $r = 0.62$. However, all of these are below the 0.7 value set for gait verification of the user.

We also validate the time it takes to run the Pearson correlation and transfer the required data via Bluetooth. We ran the Pearson correlation 100 times on the wearable security testbed. On average, the Pearson Correlation took 0.19 milliseconds to run. For the time it takes to send the validation data via Bluetooth, we tested sending the data as a batch of 100 values to two devices. We sent our test data a total of 954 times, taking an average of 5.22 milliseconds. These two results show that the time it takes to verify the user is minimal, around 6 milliseconds on average. This result is fast enough to ensure that the use of multiple different devices for

verification and storage will not slow the user down significantly. It is below the 40ms used by movies to simulate smooth motions and, thus, will not be noticed by the user.

Table 3. Pearson Correlation of Attacker Mimicing User

	Stored X	*Stored Y*	*Stored Z*
Stored X	1		
Stored Y	-0.1346	1	
Stored Z	0.20493	-0.8576	1
Mimic1 X	-0.1776	-0.3051	0.09857
Mimic1 Y	0.32715	-0.5582	0.52614
Mimic1 Z	-0.0097	0.64248	-0.5999
Mimic2 X	-0.4989	0.00445	-0.2864
Mimic2 Y	0.44001	-0.5401	0.49955
Mimic2 Z	0.03235	0.63008	-0.5612
Mimic3 X	-0.1993	-0.4968	0.32709
Mimic3 Y	0.08862	-0.5414	0.55538
Mimic3 Z	0.0272	0.42123	-0.4385
Mimic4 X	0.57266	-0.3955	0.32645
Mimic4 Y	-0.1889	0.14998	-0.061
Mimic4 Z	-0.0135	0.26993	-0.0707

8. Conclusions and Future Work

In this paper, we extend gait-based verification techniques and apply the concept of gait-based authentication to the personal fog architecture. Within the personal fog, we shift the authentication requirements from the local wearable to all connected fog nodes. This shift allows verification of multiple different gait profiles from different locations on the body and prevents an attacker from accessing personal information from stolen wearables. With the additional power assumed by the personal fog, we allow all connected fog nodes to independently verify the gait of the user using the recorded gaits of all fog nodes. We show this method is viable for wearables through testing using our wearable security testbed.

It is important to note that this method is designed to be used in small scale. A user's gait data will only be passed between their devices, never moving to the cloud. If this method is implemented on a large scale, with gait data being stored not just on a user's own devices but across multiple fog nodes outside of the user's control, the continuous monitoring of the user's data and storage in a database by a user could be exploited by an attacker to identify and target specific users based on their gait. Ideally, this method is used only on the layers of the personal fog that the user has control over and, thus, should never exceed the number of devices a user can comfortably wear.

This method has limitations. One issue is that, if an attacker is able to access the wearable without needing to then walk to a different location, they can use the existing stored gait information to authenticate and gain access to the users' private information. This outcome would likely not be an issue in most cases. However, if a user left their device somewhere, an attacker could gain access without needing to take the device to another location. This would require the user to leave all their wearables and their base station in a single location, which is unlikely, but more research is needed to prevent this possible attack.

An attacker with unlimited time to study and refine a user's gait may also be able to successfully mimic the users gait enough to fool our system. While we tried to address this situation with our own testers mimicking the stored user's gait, we did not provide our attackers with unlimited time to learn and practice the users gait. Further research is needed to discover if, given enough time, an attacker could mimic a user's gait enough to fool our system.

Another issue with this method is that gait data could be seen as medically valuable. Gait can be used to recognize health issues in a user and having their gait information stored on multiple devices could allow attackers to gain access to this health information or be used to diagnose medical conditions the user was not aware of. Should an attacker gain access to this information, either through accessing the stored data itself or through eavesdropping on the gait data as it is being passed between devices, they could gain insight into a user's health or psychological state. This problem is made worse if the gait data is passed to a third party for verification. While we have focused only on using gait data for identification in a personal fog, where all devices are owned by the user, the fog, by its very nature, can have additional nodes outside of the users control added in the future. We do not recommend using gait for identification in this case, as an attacker gaining access to a third-party node would provide them with the stored data from all users that use that node.

We would also like to extend this verification method to other devices and other methods of verification. User data is often unique enough to be used for authentication and wearables are constantly collecting data about their user. It is possible that our method of storing data for verification across multiple devices could expand to use more than just gait data.

9. References

[1] "Apple Watch Series 3 – Apple" https://www.apple.com/apple-watch-series-3/. Retrieved June 2018.

[2] S. Bouchenak, G. Chockler, H. Chockler, G. Gheorghe, N. Santos, and A. Shraer. "Verifying cloud services: present and future", SIGOPS Oper. Syst. Rev. 47, 2 (July 2013), 2013.

[3] M. Boyle, A. Klausner, D. Starobinski, A. Trachtenberg, and H. Wu. "Poster: gait-based smartphone user identification", In Proceedings of the 9th international conference on Mobile systems, applications, and services (MobiSys '11). 2011.

[4] C.C. Byers and P. Wetterwald, "Fog Computing: Distributing Data and Intelligence for Resiliencey and Scale Necessary for IoT: The Internet of Things (Ubiquity symposium)", Ubiquity 2015. 2015.

[5] G. Cola, M. Avvenuti, F. Musso, and A. Vecchio. "Gait-based authentication using a wrist-worn device", In Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS 2016). 2016

[6] S. Dey and A. Mukherjee, "Robotic SLAM – a Review from Fog Computing and Mobile Edge Computing Perspective", Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services. 2016.

[7] "Here One ™ Wireless Smart Earbuds" https://hereplus.me/. Retrieved June 2018.

[8] C.C. Ho, C. Eswaran, K.W. Ng, and J.Y. Leow. "An unobtrusive Android person verification using accelerometer based gait", In Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia (MoMM '12). 2012.

[9] "How powerful is the Apple Watch's S1 processor?" https://www.cultofmac.com/320279/how-powerful-is-the-apple-watchs-s1-processor/. Retrieved June 2018.

[10] "How to use Smart Lock to unlock your phone automatically | Android Central" https://www.androidcentral.com/smart-lock Retrieved June 2018.

[11] S. Hwang and J. Gim. "Listen to Your Footsteps: Wearable Device for Measuring Walking Quality", In Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15). 2015.

[12] A. Kattepur, H. Dohare, V. Mushunuri, H.K. Rath, and A. Simha, "Resource Constrained Offloading in Fog Computing", Proceedings of the 1st Workshop on Middleware for Edge Clouds & Cloudlets. 2016.

[13] A. Jain, L. Hong, and S. Pankanti, "Biometric Identification." Communications of the ACM 43.2 (2000): 90-98.

[14] M. Muaaz and R. Mayrhofer. "Accelerometer based Gait Recognition using Adapted Gaussian Mixture Models", In Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media (MoMM '16). 2016.

[15] M. Muaaz and R. Mayrhofer. "Orientation Independent Cell Phone Based Gait Authentication", In Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia (MoMM '14). 2014.

[16] R. Mayer, H. Gupta, E. Saurez, and U. Ramachandran, "The Fog Makes Sense: Enabling Social Sensing Services with Limited Internet Connectivity", Proceedings of the 2nd International Workshop on Social Sensing (SocialSens'17). 2017

[17] "Nymi | Always On Authentication™" https://nymi.com/. Retrieved August 2018.

[18] I. Papavasileiou, S. Smith, J. Bi, and S. Han. "Gait-based continuous authentication using multimodal learning", In Proceedings of the Second IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE '17). 2017.

[19] "Samsung Gear: Smartwatches & Fitness Trackers | Samsung US" https://www.samsung.com/us/mobile/wearables/. Retrieved June 2018.

[20] S. Sharma, R. Tiwari, A. Shukla, and V. Singh. "Frontal view gait based recognition using PCA", In Proceedings of the International Conference on Advances in Computing and Artificial Intelligence (ACAI '11). 2011.

[21] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A Hierarchial Distributed Fog Computing Architecture for Big Data Analysis in Smart Cities", Proceedings of the ASE BigData & SocialInformatics 2015. 2015.

[22] C. Walter, I. Riley, and R.F. Gamble. "Securing Wearables through the Creation of a Personal Fog", In Proceedings of the 51st Hawaii International Conference on System Sciences. 2018.

[23] W. Xu, Y. Shen, Y. Zhang, N. Bergmann, and W. Hu. 2017. "Gait-Watch: A Context-aware Authentication System for Smart Watch Based on Gait Recognition", In Proceedings of the Second International Conference on

Internet-of-Things Design and Implementation (IoTDI '17). 2017.

[24] W. Xu, C. Javali, G. Revadigar, C. Luo, N. Bergmann, and W. Hu. "Gait-Key: A Gait-Based Shared Secret Key Generation Protocol for Wearable Devices", ACM Trans. Sen. Netw. 13, 1, Article 6 (January 2017).