Reconstructing what you said: Text Inference Using Smartphone Motion

Duncan Hodges, Oliver Buckley

Abstract—Smartphones and tablets are becoming ubiquitous within our connected lives and as a result these devices are increasingly being used for more and more sensitive applications, such as banking. The security of the information within these sensitive applications is managed through a variety of different processes, all of which minimise the exposure of this sensitive information to other potentially malicious applications. This paper documents experiments with the 'zero-permission' motion sensors on the device as a side-channel for inferring the text typed into a sensitive application. These sensors are freely accessible without the phone user having to give permission. The research was able to, on average, identify nearly 30% of typed bigrams from unseen words, using a very small volume of training data, which was less than the size of a tweet. Given the natural redundancy in language this performance is often enough to understand the phrase being typed. We found that large devices were typically more vulnerable, as were users who held the device in one hand whilst typing with fingers. Of those bigrams which were not correctly identified over 60% of the errors involved the space bar and nearly half of the errors are within two keys on the keyboard.

1 Introduction

The use of mobile devices, whether smartphones or tablets, has become ubiquitous with our hyper-connected lives [1]. They are now used for not only communicating with friends and family but also for performing all sorts of tasks ranging from accessing the internet through to more sensitive applications such as shopping and banking. Many popular sites report that access from mobile devices is more far common than 'traditional' laptop / desktop access. This shift in the use of mobile devices from a personal communication tool to a personal 'productivity' tool has increased the amount of potentially sensitive material and activity performed on them. These smartphones have become increasingly personal and how we trust others and share these phones has become increasingly complicated [2].

As these devices now hold such sensitive information it is more important than ever to be able to secure them and much research has been performed on the permissions model governing them (e.g. [3], [4], [5], [6]). This permissions model, in addition to the file storage model [7], are the key mechanisms by which the Android devices attempt to protect sensitive information in one application from other, potentially malicious, applications. The research presented in this paper explores one particular way to bypass this security model such that one application can 'read' the data being typed in another application. In essence this creates a keylogger capable of extracting sensitive data input into other applications.

In this paper, we discuss an experiment using Android smartphones and tablets that demonstrates it is possible to infer the key presses on any Android smartphone or tablet purely from the motion sensors. The motion sensors are freely accessible to applications on the device, without express permission being sought from the user. This paper continues in Section 2 with a discussion of the academic background to this research, Section 3 then discusses an experiment to explore typing on soft-keyboards. Section 4 discusses the analysis of the experimental data, whilst Section 5 covers the implications for mobile users, app developers and device manufacturers. Finally the conclusions of the paper are explored in Section 6.

- *O. Buckley is with the School of Computing Sciences within the University of East Anglia.*
- *D. Hodges is with the Centre for Electronic Warfare, Information and Cyber, Cranfield University, Defence Academy of the United Kingdom, Shrivenham, Swindon, SN6 8LA.*
  E-mail: o.buckley@uea.ac.uk, d.hodges@cranfield.ac.uk

Manuscript received Month Day, Year; revised Month Day, Year.

2 Background

Digital connectivity is becoming more increasingly intertwined with our daily lives and as a result the tasks that are performed on smartphones and tablets have become more personal and we have become inseparable from our smartphones [8], 40% of smartphone users describe them as *'something they could not live without'* [9]. This degree of dependence is twinned with an increase in the breadth and sensitivity of tasks performed on these smartphones with 43% looking for information about jobs, 40% accessing Government services or information, 62% looking up information about health conditions, 44% looking up real estate listings or information about a place to live and 57% access online banking services [9].

The standard approach to security on personal devices is through a permissions-based model, which relies on the users having the ability to perform a relatively complex risk-based security decision in order to allow an application to access potentially sensitive information (such as the address book) or capability (such as location sensors). This complicated model has been shown to be difficult for most users to manage. This can be either because users are unable, or unprepared, to fully realise the risk associated with granting permissions to an app [6] or because apps are 'over-priviliged' meaning they request greater permissions than are required to perform their function [10]. In addition to

this permissions model, application sandboxing attempts to limit the effect of an untrusted or malicious application. This sandboxing attempts to ensure that an app cannot access information within other, potentially sensitive, applications (whether during execution or data 'at-rest'). There have been several pieces of malware on the Google Play store that have been downloaded a significant number of times that either fraudulently send SMS or perform other malicious acts. Researchers have identified eleven such applications which have each been installed over 5 million times [11].

In this research we are concerned with the side-channels that could potentially allow information to leak from a sensitive application to a malicious application. There has been research which has considered the electromagnetic and power profiles of the device in order to infer sensitive information from within applications [12]. We particularly focus on how the phone moves in order to infer information from one sensitive application.

The sensors in smartphones have been used to good effect to infer a wide range of information about an individual solely based on the way that they interact with the smartphone's touchscreen, for example inferring the length of the user's thumb [13] and as a result estimating their height or being used infer the user's gender [14]. They can also be used to infer the user's gait patterns [15], the activity being performed [16] even location and travel routes [17], [18]. The accelerometers have also been used to infer information outside the context of the device, effectively using it as a network enabled sensor [19], [20].

Motion sensors within smartphones have previously been used to attempt to infer a user's keystrokes with promising results. TouchLogger [21] was a smartphone application designed to infer the keystrokes on a soft keyboard based solely on the vibrations recorded by the smartphone's motion sensors. The research was capable of successfully inferring more than 70% of the keys that were typed using only the device's accelerometer. However, the work focused specifically on inferring the keystrokes from a soft keyboard that contained only numbers. The work that we present in this paper will look to infer the keystrokes of an individual that use a standard soft keyboard. Similarly, Xu et al. [22] present TapLogger, an approach that looks to infer an individual's taps on a numeric keyboard using a smartphone's accelerometer and gyroscope. This work differs from our own in that it focuses on identifying single taps, which are more susceptible to distortion by linear drift. Our approach looks to identify pairs of keystrokes, or bigrams, which provides a much more robust identification method that is less affected by linear drift. The work of Aviv et al. [23] builds on the idea of PIN identification using motion sensors. However, where previous work had focused on single taps, their work introduces the notion of recognising and inferring the swipe gestures of a user's PIN.

More recently Shen et al. [24] have built in the approach taken by TapLogger [21] with taps being recorded with the accelerometer with the gyroscope and magnetometer used to infer the positions of the taps. The authors carried out an empirical study in line with previous studies, with 30 participants using a range of screen sizes, data sizes and sampling rates. Their work focused on identifying key presses on a numeric keypad where they found they could detect when the key presses occurred 100% of the time and were able to correctly identify the key that had been pressed 80% of the time 'in some cases'. The accuracy of key detection ranged from 71.4% to 83.9% depending on the conditions. Again, this work differs from the work that we present in that it focuses solely on detecting key presses on a numeric keyboard as opposed to our own work that uses the an alphanumeric keyboard. Additionally, Shen et al. [24] aim to identify single taps, as with the work of Xu et al. [22], which can lead to distortion.

Other work has focused on password compromise, for example, Owusu et al. [25] uses a smartphone's accelerometer to infer the characters, both letters and numbers, contained within a user's password, although with a relatively small set of only four participants. The work was capable of extracting the 6 character passwords in around 4.5 attempts.

The work of Miluzzo et al. [26] on TapPrints builds on this idea of keystroke identification by looking to determine the location of the tap on a smartphone screen. This is used initially to understand the icons that may have been tapped, and so the applications that were launched. This work then takes this concept further to try and identify individual keystrokes on a virtual keyboard. This work focusses on single taps, which again will perhaps provide a less robust result, that can be susceptible to linear drift. The results suggest that there is an 80% accuracy in predicting an individual letter which is in line with our own results of around 81% accuracy in bigram prediction. However, the sample size of only 10 volunteers is notably less than the 53 participants in our own work but the dataset of around 40,000 keystrokes is much larger than our own. Our work collects 138 characters per participant, which is less than the size of a Tweet, as we are focused on the minimum amount of data that might be needed to correctly identify a user's keystrokes.

The majority of the current work in this area relies solely on the use of smartphone motion sensors, however, the work of Narain et al. [27] builds on this idea to incorporate the stereoscopic microphones on an Android smartphone. This work developed a method that used both the gyroscope and stereoscopic microphones on a smartphone and that was around 90% accurate in its predictions. This work was conducted with only seven participants and was limited to three devices (Samsung S2, Samsung Tab 8 and HTC One), whereas our work allowed for any compatible Android device.

The majority of the work has focused only on the use of accelerometer readings, in contrast to our own work, which includes analysis of rotational data using the smartphone's gyroscope. When a phone is being used by an individual it tends to be held in a hand that is either unsupported or with the wrists resting on a surface, if the device is being held in two hands with the thumbs for typing it tends to be held loosely and tilted in the palms in order that the relevant keys are closer to the thumb. If a device is held in one hand the same phenomena occurs however the aim tends to be to reduce the amount the 'pecking' digit has to move. Whilst these movements are relatively subtle they are observable both by the human eye and by the smartphone's sensors.

We anticipate this behaviour is repeatable based on the concept of motor learning, the process by which individuals

acquire or develop skills, through experience. Fitts and Posner [28] suggest a three-stage model of motor learning where a skill will become more 'automatic' as familiarity increases, this results in a performance requiring little or no conscious thought. This lack of concious thought has led to typing on traditional keyboards being as distinguishing as an individual's handwriting or signature, since there are similar governing neurophysiological mechanisms [29], [30]. There is no fundamental reason why this same process of motor 'memory' should not also be the case in governing the use soft-keyboards on devices and resulting in a repeatable unconscious motor behaviour.

3 METHOD

The initial pilot experiments in this research used one device (a Google Nexus 5X) upon which all the experiments were performed, this demonstrated the correct identification of approximately two-thirds of pairs of keystrokes that had been seen before and around a quarter of pairs of key presses not seen before [31]. This work expands on the initial research by allowing the results to be gathered from a range of phones and tablets and by individuals not being directly supervised by the researcher. It is very important that a participant is able to use their own phone, our initial study relied on using a device which they may not have seen before, this will effect the natural way in which the device is used and may have under-represented the repeatability and hence the transition of the model from training to test phases. This paper also introduces different approaches to building and matching the model such as Dynamic Time Warping, these whilst more computationally expensive are expected to produce significantly improved results.

In order to ensure that the application was available to a wider audience it was submitted to both the Google Play store and the Amazon App store, the main App Stores for the Android ecosystem. A number of previous researchers have taken similar approaches to delivering research applications to a wide range of users and devices throughout the world, most notably the *Device Analyser* application [32].

The application was compatible with over 13,000 devices on the Google Play Store, and followed a very simple flow. An initial Activity provided the consent and participant information sheet whilst also checking that the device had the relevant sensors available. Following this participants were asked a number of demographic questions including: their age, handedness, how long they had owned the phone or tablet and how comfortable they were using the keyboard. The next Activity asked how the participant held the phone whilst typing (whether in the left hand, right hand or both hands) and what the participant uses to press the keys (just fingers, just thumbs or both fingers and thumbs). There were no requirements placed on the orientation of the device, with the participants free to use the device in the way in which they felt most comfortable. We found that the orientation of the device did not impact the results or accuracy of the research. The participant is then asked to type a paragraph of fixed text twice, this formed 'experiment 1' and 'experiment 2', the text was calculated to explore the full breadth of the keyboard whilst being short enough to be approximately the length of a tweet (132 characters):

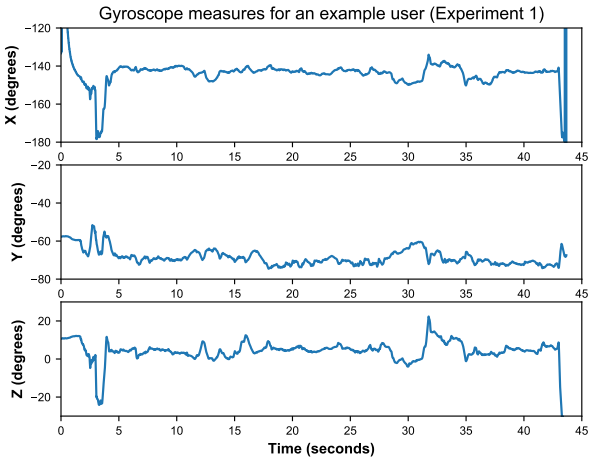

Fig. 1. Example timeseries captured from the gyroscope.

fly me to the moon and let me play among the stars our freedom of speech is freedom or death we have got to fight the powers that be

The participants were then asked to type a random set of English words that were constructed from the bigrams (two character substrings) in the fixed text, this formed 'experiment 3'. Finally participants were asked how easy they found the task and how distracted they were. The application then launched their email program and attached the data files with the final consent form and the participants were asked to re-read the consent form before sending the data to the researchers. During the study the device recorded the outputs from the gyroscope sensor and the accelerometer. These time-series along with the key down and key up times from the keyboard form the main experimental data. During the study it was assumed that all participants were stationary, although this was no explicitly specified. It is entirely possible that the accuracy of the results could be altered in different scenarios, for example when walking. The research presented here does not assume that this attack would be successful in all scenarios. For this study auto-complete was disabled in order to rely purely on the observed typing behaviour, and at this stage 'swipe' keyboards were disabled so the participants purely used the standard action of pressing individual keys.

The sensors on smartphones are generally small devices and often subject to drift, by using the rotation from one keypress to another keypress to characterise the two keypresses this method is only affected by the non-linear component of changes in the sensor performance. Should we attempt to identify individual keypresses using the gyroscope and accelerometer we would be affected by both the linear and non-linear changes in the sensor performance. It may be possible to use tri-grams or higher orders of n-gram, however the volume and diversity of training data required becomes significant.

An example of the data gathered as part of the experiment is shown in Figures 1 and 2 for one example user.

Participants were recruited across traditional social media channels using snowball sampling [33] and also publi-

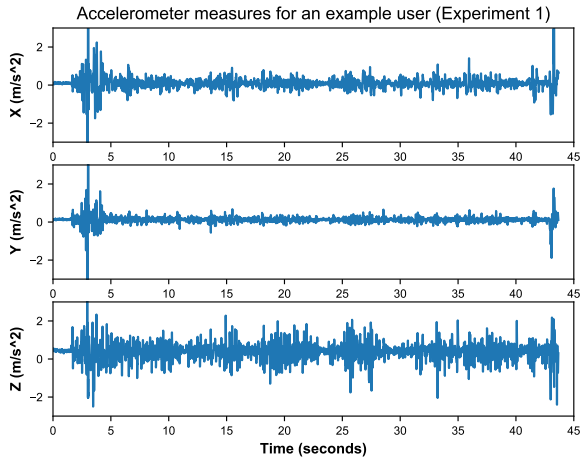

Fig. 2. Example timeseries captured from the accelerometer.

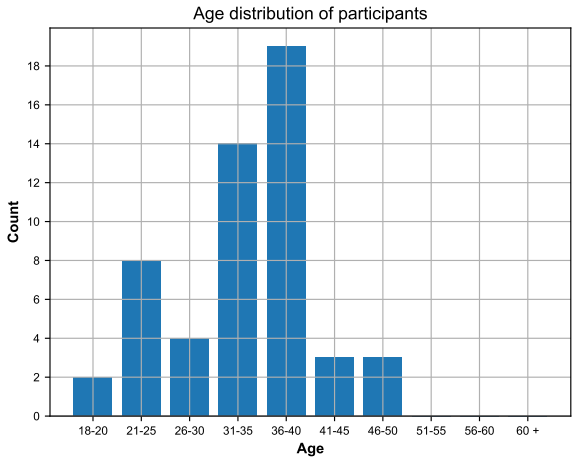

Fig. 3. Distribution of the age of the study participants.

cising the study through participant recruitment channels of Reddit and other websites. This wider reach attempted to reach as broad a range of smartphone and tablet users as possible. The study recruited 53 participants, all of whom were over the age of 18.

4 Analysis

The cohort was recruited with the age distribution shown in Figure 3, as can be seen most participants fall into the 20—40 age bracket with the largest group of participants being in their thirties. The participants generally had their device for less than 18 months, as shown in Figure 4 and generally considered themselves at least comfortable with the soft-keyboard as seen in Figure 5. We found no correlation between the participants' age and how comfortable they were with the soft-keyboard, we also found no correlation between how long the participant had owned the device and how comfortable they considered themselves to be. However, as could have been predicted we did find a correlation between how comfortable a participant was with a soft keyboard and how easy they found the task (when asked after the experiments), a Pearson's Chi squared test resulted in a Chi squared statistic of 15.877 (p-value of 0.01443).

Through distributing the application across the App Stores the aim of the study was to explore the performance not only across a variety of individuals but also a variety of devices. The participants used a range of phones and tablets from a number of different manufacturers, see Figure 6, this distribution is well aligned with that we would expect to see in the Android marketplace, for example as shown in [34] with the exception of Lenovo devices potentially being under-represented within our sample.

As we would expect the devices used in this study had a range of sizes and screen resolutions the resolutions were grouped into the descriptive terms used by the Android OS which are defined as MDPI ($x \leq 200$), HDPI ($200 \leq x < 280$), XHDPI ($280 \leq x < 400$), XXHDPI ($400 \leq x < 560$) and XXXHDPI ($560 \leq x$).

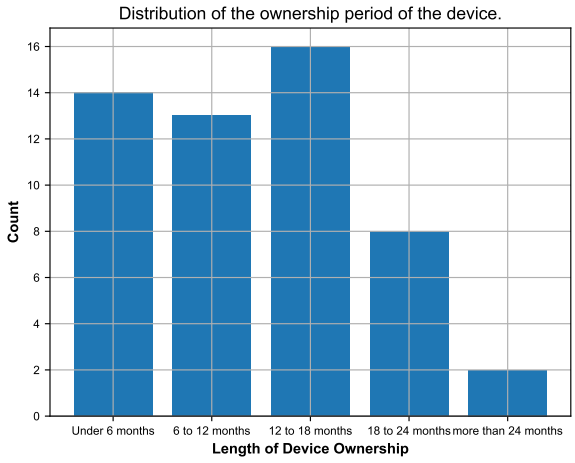

Fig. 4. Distribution of the length of time participants have had the device.

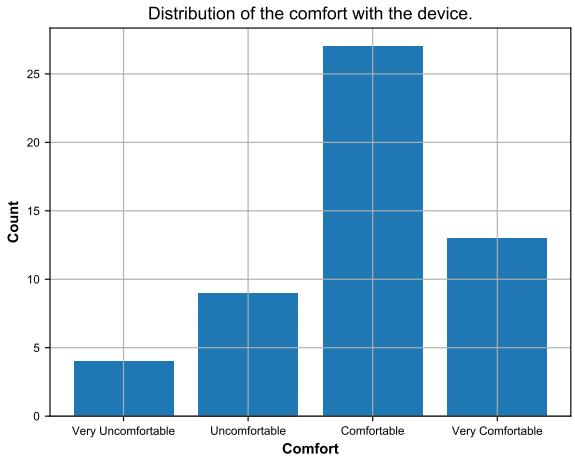

Fig. 5. How comfortable participants were with their device keyboards.

The physical size was calculated using the resolution and the screen size in pixels — it should be noted that this is only

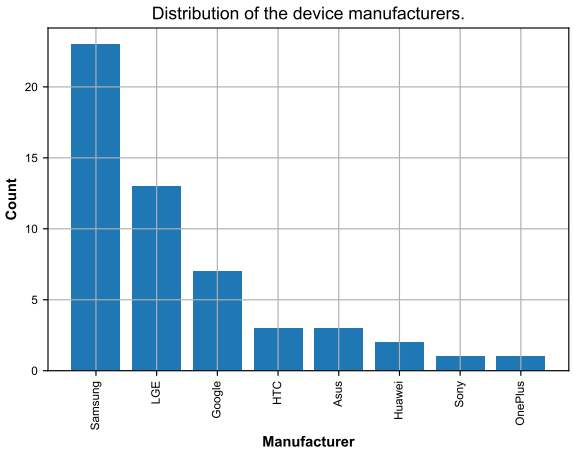

Fig. 6. The device manufacturers used by participants in the study.

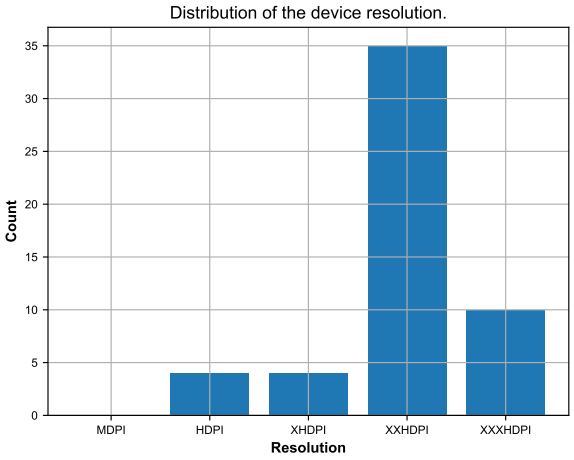

Fig. 7. The resolution of devices used by participants in the study.

the touchscreen size and does not include the size of the bezel or external housing. The distribution of the device resolution and sizes are shown in Figures 7 and 8 respectively, these show that most devices had higher screen resolutions (consistent with most devices being under 18 months old) and generally between 4.5 and 6 inches (consistent with smartphone or small 'phablet' sizes) in addition to a number of 7 and 10 inch tablets.

The Android ecosystem divides device sizes into *Small* ($\leq$ 3.5 Inch), *Normal* ($3.5 \leq x < 5$), *Large* ($5 \leq x < 7$) and *XLarge* ($7 \leq x$). Interestingly whilst there are indications of a relationship between the device size and the comfort level with users of Large and XLarge devices generally being less comfortable than those of normal sized, a Pearson's Chi-sqaured test resulted in a statistic of 12.17 and a p-value approaching significance at 0.058. Approximately three quarters of those participants with XLarge devices reported they were uncomfortable with the soft-keyboard.

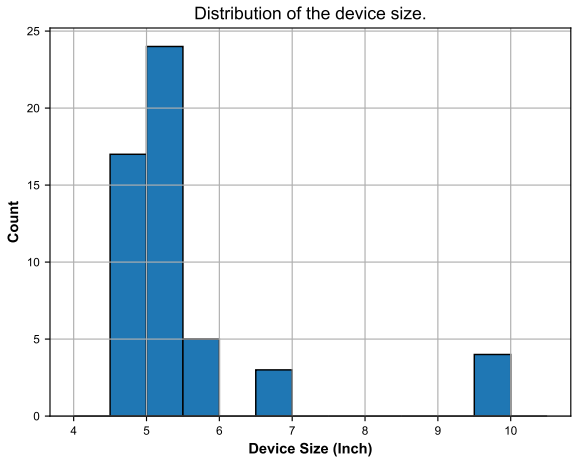

Fig. 8. The size of devices used by participants in the study.

4.1 Identification of Key Presses

As described previously the approach that we take in this research is the use of the motion of the device in order to identify the bigram that was typed. We must first identify the timestamps of the two key presses forming the bigram and then use the rotation between those timestamps to identify the letters forming the bigram.

The first step in this approach is to be able to identify the key presses, for this we use the accelerometer on the device. We annotated a key press when the magnitude of the acceleration component exceeded a threshold, this threshold was derived using the key press data gathered in the training phase of experiment 1. This threshold was not constant across the study, and was a result of the training phase using experiment 1 and hence was a function of both the individual's typing and the device.

The magnitude of the acceleration was found to be the most discriminative for the identification of the key presses, for some devices the acceleration in the 'z' direction (i.e. 'into' the device) was also discriminative, however we found that for some larger devices the performance was worse than using the magnitude.

The magnitude of the accelerometer measure for an example user is shown in Figure 9. This shows the optimum threshold for this particular user working as derived from the training data associated with experiment 1.

The Relative Operating Characteristics (ROC) [35], [36] of this detection process is shown in Figure 10, as can be seen the performance of the detector is, in general, good.

The Area Under Curve (AUC) metric is the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [37]. A boxplot of this metric for each participant across the experiments are shown in Figure 11, the average performance being 0.885, in effect correctly identifying the time at which key presses occur approximately nine out of ten times.

4.2 Identification of Bigrams

In this research we are primarily interested in identifying the potential side-channel available through the movement of

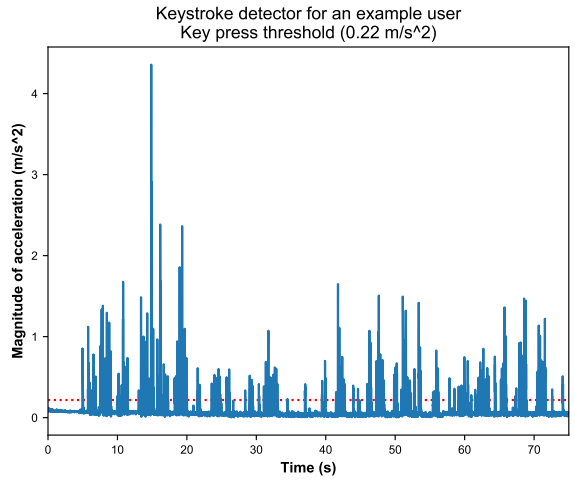

Fig. 9. An example of the threshold used for the keypress detection.

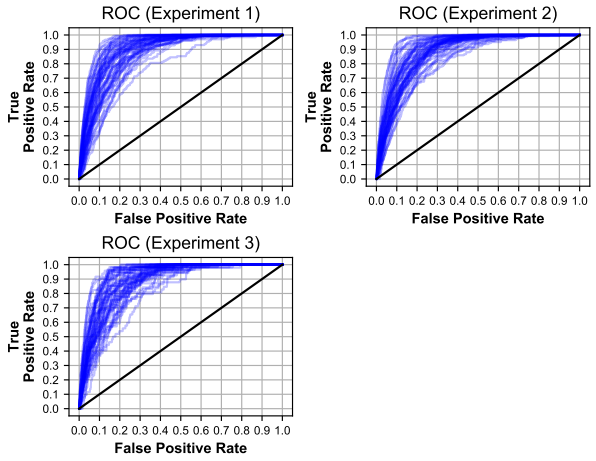

Fig. 10. The Relative Operating Characteristic of a simple threshold detector for detecting key presses.

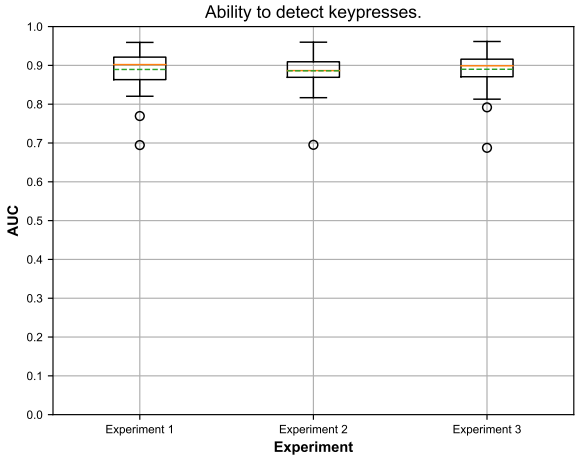

Fig. 11. The AUC metric for detecting key presses across the three experiments.

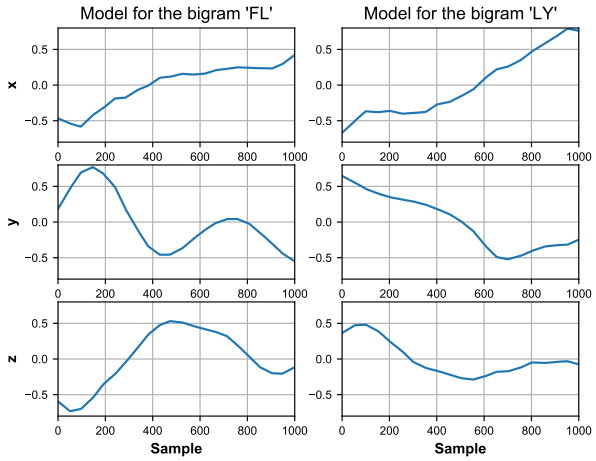

Fig. 12. The example model components for the bigrams ‘fl’ and ‘ly’.

the device and as such the initial mechanism that was used to identify the bigrams used a very simple model. More complex models are explored in Section 4.5.

Initially this process trained a model for an individual participant on the recordings from experiment 1, this model was then validated using experiment 2 and 3. Using experiment 1 as training data we first identify the rotation vectors in x, y and z dimensions that are associated with each bigram, i.e. the time-series from the rotation sensor from the first key-down until the second key-up. These time-series were resampled to a common number of regular samples (in this case 1,000 per bigram) to remove any linear variation in the flight time between key presses. Each vector component is then normalised by removing the mean — this was found to reduce the effect from previous bigrams since the model now simply encodes the change in rotation rather than being dependent on the starting orientation of the phone. Where bigrams had multiple occurrences in the training data the vectors were averaged.

An example of the model components for one user is shown in Figure 12. There are some similarities in the x and y rotation captured in the model, however model components capturing the rotation in the z direction are the inverse of each other. Noting the positions of the ’f’, ’l’ and ’y’ keys on a keyboard we can clearly see the left-to-right motion captured for the ’fl’ bigram and then right-to-left motion captured for the ’ly’ bigram in the rotation in the z dimension.

This model was then verified against the data from experiment 2 and 3, this process identified key presses using the approach described in Section 4.1, the rotation vectors were then extracted between these timepoints. These rotation vectors were then resampled and the bigram in the model with the lowest RMS error was deemed to be the corresponding bigram. This very simple ‘naive model’ treats each bigram as completely independent meaning that a poor prediction in one bigram is not propagated to those adjoining bigrams.

We can create a more realistic model since, logically, the first letter of a bigram must be the last letter of the preceding bigram. This ‘bigram model’ takes the error across

TABLE 1
Example bigram scores from Naive model.

Position 1		Position 2		Position 3	
Bigram	RMS Err.	Bigram	RMS Err.	Bigram	RMS Err.
|ab|	0.01	|bc|	0.03	|t |	0.04
|db|	0.05	|ec|	0.06	|c |	0.05
|ae|	0.07	|bu|	0.10	|um|	0.15

TABLE 2
Resulting Bigram model sentences from the example Naive model output shown in Table 1.

Sentence	Total Error
|abc |	$0.01 + 0.03 + 0.05 = 0.09$
|dbc |	$0.05 + 0.03 + 0.05 = 0.13$
|aec |	$0.07 + 0.06 + 0.05 = 0.18$
|abum|	$0.01 + 0.10 + 0.15 = 0.26$
|dbum|	$0.05 + 0.10 + 0.15 = 0.30$

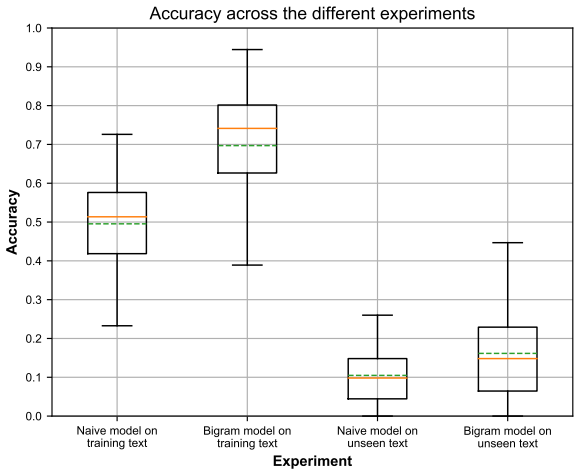

Fig. 13. The accuracy across the two different experiments using the two different methods.

all predictions of individual bigrams and attempts to create the sentence with the lowest overall error which obeys the logical rule that a bigram must start with the preceding bigram's end letter. Initially this method uses the naive predictions which give an error measure of a particular bigram in a particular position in the sentence, in the naive model we simply take the bigram in a particular position which produces the minimum error. In this 'bigram model' we take the first bigram in the first position, then consider all the bigrams in the second position which begin with the end letter of the bigram in the first position. This represents the possible two bigram combinations which are logically possible, the error term of these bigram pairs is then the sum of the individual error terms. This process continues until the all logically possible combinations beginning with the most likely start bigram have been calculated, the process then moves onto the next possible start bigram. Hence we calculate the error term for all logically possible combinations of bigrams, to manage the computational challenge of this at every level we only retain the million combinations that have the lowest error. Note, this logical assumption does not consider the use of language or the fact that some bigrams are more likely in some languages. A trivial example of this process is shown in Tables 1 and 2.

The ability of these two models, the 'naive model' and the 'bigram model', is shown in Figure 13, the accuracy is measured as the number of correctly guessed bigrams normalised to the number of bigrams. Average performances of 49.5% and 69.7% were displayed from text that has been seen before and 10.5% and 16.1% for new text. While this performance may appear low, note that this performance is from a single piece of training text less than the size of a tweet (132 characters) using a trivially simple model, and for some participants this performance is significantly higher.

We found that the errors were not randomly distributed across the bigrams. The ten most common confusions are shown in Table 3, this shows the normalised count of confusions, i.e. the number of times bigrams are incorrectly guessed normalised by the number of times the bigram appears in the text. As can be seen the most common mistakes are generally close on the keyboard.

TABLE 3
The most commonly confused bigrams.

Truth	Predicted	Normalised Count
|s |	|e |	10.7
|h |	|b |	9.0
|le|	|me|	8.0
|h |	|n |	8.0
|h |	|om|	8.0
|t |	|r |	7.8
|e |	|d |	7.7
|at|	|ar|	7.5
|t |	|e |	7.2
|h |	|to|	7.0

It is noteworthy that 62.4% of the errors are caused by bigrams that involve the space-bar, the space-bar is significantly wider than other keys this means the target area for the digit is much larger than for other keys. This larger target means that the user is less constrained as they move from key-to-key and hence the behaviour is less repeatable. For example, the error with the highest normalised count was mistaking the bigram `s` followed by `SPACE` with the bigram `e` followed by `SPACE`. The keys `s` and `e` are adjacent on a keyboard with the space bar underneath both and hence the rotation of the phone or tablet will be similar between the two bigrams; this means a user is likely to use a similar movement to transition from one keypress to another. Should the start keys be further apart the center of the rotation of the device is likely to involve a more discriminatory component in the 'roll' of the device.

Figure 14 shows the Cumulative Distribution Function (CDF) of the distance from the correct key to the key that was in error. From this we can see when an error occurs that around 40% of the first letter of the bigram when in error was within 2 keys of the correct letter and around 50% of the second letters when in error were within 2 keys.

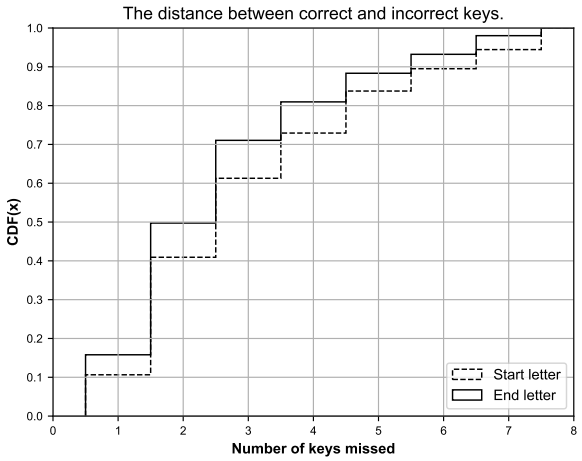

Fig. 14. The distance between the correct and predicted key for the bigrams in error.

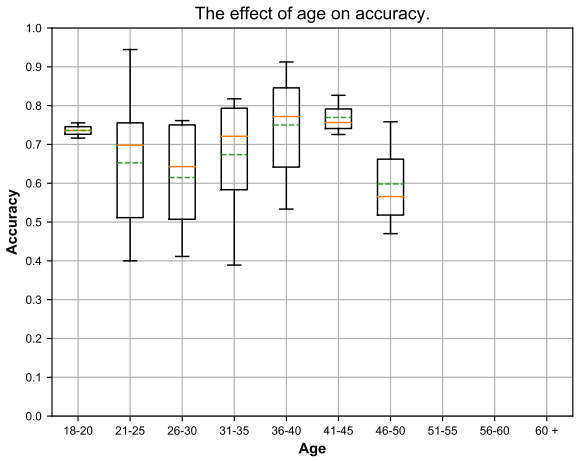

Fig. 15. The accuracy from different age participants.

4.3 Factors affecting the accuracy

In our initial work on this topic we explored a small subset of factors that could influence the accuracy of the prediction method including whether fingers or thumbs were used to type on the soft-keyboard, how comfortable the participant was with the keyboard and the median flight-time per participant (that is the time from a key-up to the next key-down) and we found no effect from these factors [31]. Since this initial study only used one smartphone (a Nexus 5X) we were unable to explore the effect the device itself had on the accuracy, and since most participants were not 'owners' of the phone they had less opportunity to become comfortable with the device and keyboard (particularly salient for non-Android users).

Understanding the factors that modulate the accuracy of the predictions is essential in fully understanding the phenomena and whether this form of information leakage is more of a threat to particular individuals or devices in the Android ecosystem.

Initially we explored the effect of the age on the accuracy of the most successful prediction experiment (that of using the bigram approach on text that had been seen previously). A boxplot of the effect of age is shown in Figure 15, from this plot it appears that there is no evidence that performance is modulated by the age of the participants, this is confirmed with a Pearson's Chi-squared test (test statistic of 0.127 and a p-value of 0.364)

The length of ownership is also of interest, again there was no correlation between how long the device has been owned and the accuracy — a Pearson's Chi-squared test confirms this (test statistic of -0.069 and a p-value of 0.623).

Mechanically the physical size of the phone or device will have some effect on how it is used, for this analysis we break the devices down using the Android 'size' metric. The accuracy of the approach broken down by the size is shown in Figure 16. As can be seen the performance on 'XLarge' (i.e. those over 7 inches) devices appears to be higher than those 'Normal' or 'Large' size. This is confirmed with two-sided Kolmogorov-Smirnov tests as shown in Table 4 which shows that participants using a device over 7 inches in size in general resulted in an increase in accuracy over those using smaller phones, albeit at a significance level of 8.7%. It is notable from inspection of the model parameters that the rotation vectors associated with larger devices tend to be larger than the parameters associated with smaller phones. This indicates the mechanical rotation of larger tablets during typing is greater.

The final question, before the study itself, asked the participant how comfortable they were using the soft-keyboard, the Kernel Density Estimation (KDE) is shown in Figure 17.

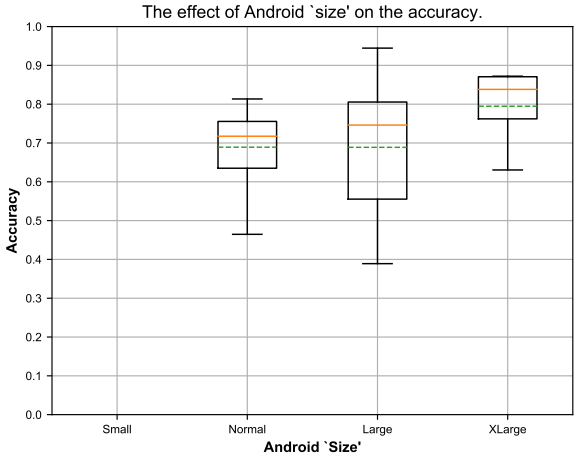

Fig. 16. The effect of the device size on the accuracy.

TABLE 4
The test statistic from two-sided Kolmogorov-Smirnov tests for the size of the device, p-values are denoted by * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$..

	Small	Normal	Large	XLarge
Small	No data			
Normal		-	0.250	0.632 *
Large		-	-	0.500
XLarge		-	-	-

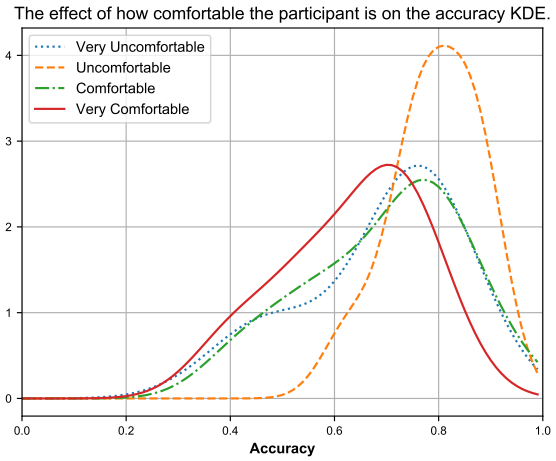

Fig. 17. The effect of comfort with the device on the accuracy of the predictions.

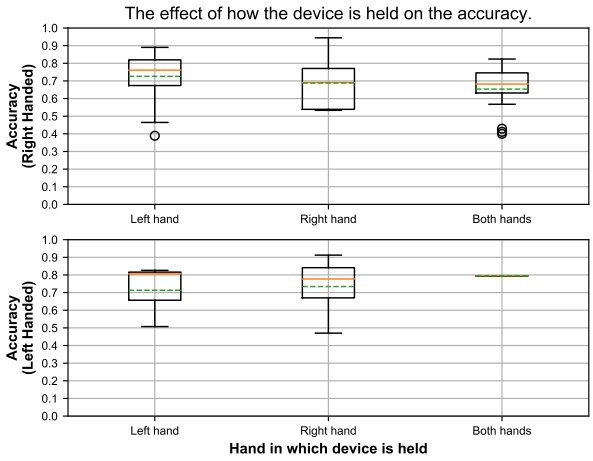

Fig. 18. The effect of how the device is held on the accuracy of the predictions.

We were unable to find a significant correlation between the accuracies across all the factors (from Very Uncomfortable to Very Comfortable), however the Uncomfortable factor was significantly higher accuracy than the very comfortable as shown in Table 5. Since the majority of the Xlarge devices were associated with users who were Uncomfortable with a smartphone keyboard it is difficult to isolate these factors.

The comfort level explored in Figure 17 is a self-declared comfort level, potentially of more interest is the actual observed typing speed as this is likely to be a better indicator of the skill level. The median flight-time (the time between releasing one key and pressing the next) of an individual's performance was calculated and no significant relationship was found with respect to the accuracy.

In addition to these factors surrounding the participants, the accuracy may also be a factor of how the device is held and used. The accuracy of the prediction when broken down by handedness and the hand in which the device is held is shown in Figure 18. Across these six groups (three different approaches to holding the phone or tablet and right- and left-handedness) there was no statistically significant difference between the performance of the prediction system. Two of the groups associated with left-handed participants who hold their device in their right hand or both hands were potentially under-represented, however given the number of participants this potentially is an uncommon use pattern.

In combination with how the device is held the digit that is used to press the keys may also have an effect on the accuracy, the digit could be just the fingers (typically with the device held in the other hand), just the thumbs (typically with it held in both hands) or a mixture of the two. The KDE covering the accuracy of the prediction as a function of these different use patterns is shown in Figure 19.

In this case there does appear to be some variation between the different use patterns, two-sided Kolmogorov-Smirnov tests were performed across the three factors and the results are shown in Table 6. As can be seen there is a statistical difference between those who type with just their fingers and just their thumbs (the relatively uncommon condition where people use both does show some reduction in performance it does not achieve statistical significance). This reduction in performance is explained by directly observing the traces associated from the devices - in general when the device is held in both hands and the thumbs used for the key presses the magnitude of the rotation vectors tend to be smaller and hence the discriminatory power is reduced.

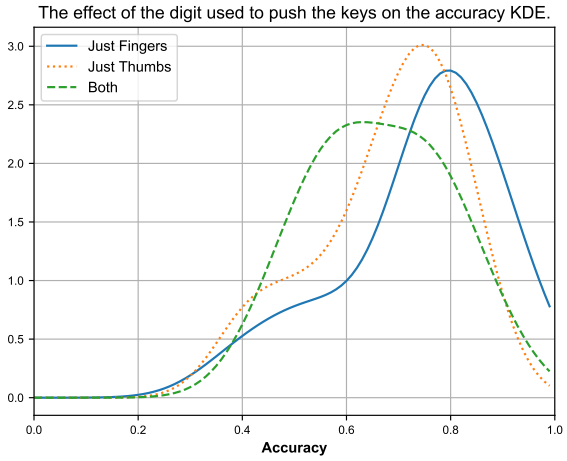

Fig. 19. The effect of how key is pressed on the accuracy of the predictions.

As previously described the length of ownership has no effect on the accuracy of the process, however this measure is linked to the age of the device, as part of the data capture the manufacturer and model of the phone or tablet is captured and this was used to calculate the age, we could find no evidence of a relationship between the age and the accuracy. After the experiment the participants were asked how focused they were during the task and also how easy they found the task, this is particularly interesting given that predictive text was disabled so it was important to gather how easy the participant found the experiment. However, we found no correlation between accuracy and the how easy

TABLE 5
The test statistic from two-sided Kolmogorov-Smirnov tests for the effect of how comfortable the participant was with the soft-keyboard, p-values are denoted by * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

	Very Uncomfortable	Uncomfortable	Comfortable	Very Comfortable
Very Uncomfortable	-	0.343	0.343	0.596
Uncomfortable	-	-	0.407	0.658 ***
Comfortable	-	-	-	0.362
Very Comfortable	-	-	-	-

TABLE 6
The test statistic from two-sided Kolmogorov-Smirnov tests for the different ways in which a keypress occurs, p-values are denoted by * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

	Just Fingers	Just Thumbs	Both
Just Fingers	-	0.380 **	0.444 *
Just Thumbs	-	-	0.325
Both	-	-	-

TABLE 7
The average accuracy using a polynomial approximation for the model across the different experiments.

Experiment	Accuracy (%)	Performance change (%)
Naive training	44.8 %	-9.42 %
Bigram training	66.1 %	-5.18 %
Naive unseen	9.90 %	-5.70 %
Bigram unseen	15.2 %	-5.54 %

the participant found the task or how distracted they were.

Having analysed the effects of these various factors in isolation we attempted to build linear models for the accuracy given the combination of these factors. No model was able to explain more than 22% of the variance, as measured by the adjusted R-squared value [38] indicating that none of the factors captured in this study are predictive of the accuracy, other than those discussed previously.

4.4 Stability in prediction over time

As discussed in the introduction the sensors involved in smartphones and tablets are small, sensitive and subject to drift over time. The experiment was re-run with a 10% sample of the participants - this sample was taken to ensure a spread of devices. This experiment ran 6 months after the initial experiment and was aimed at exploring the effect of changes in sensor behaviour over time.

For all users the performance was within 5% of the initial experiment (with an average difference of 0.2 %) indicating similar performance levels. More interestingly, further analysis used the training data from the first run to build a model that was then used to predict the keypresses from data acquired 6 months later. This resulted in an average performance decrease of 8.1%.

This decrease can be attributed to the non-linear drift in the sensor, as we are using the relative change in the phone from one keypress to the next in order to infer the pair of keypresses we are unaffected by any linear drift.

4.5 Complex models

To this point we have demonstrated a very simple approach to modelling the side-channel leakage, through building a model from the recorded time series during the training phase (experiment 1). By visual inspection the models representing the bigrams were generally smooth curves. In this example the full time-series that was used to encapsulate the model for each bigram could be replaced with the coefficients from a polynomial function. A fourth-order polynomial was found to be the minimum order of polynomial that was able to fit the data well.

By simplifying the model using this polynomial approximation it was possible to reduce the size of the model by 250 times - particularly important if the model is to be stored on the device and used in real-time. This simplification leads to a small decrease of the average performance across the experiments of approximately 5 to 6%. In some cases, particularly models that performed poorly, this approach actually improved the accuracy by reducing the noise component within the training phase of the model creation. Table 7 compares the performance of the full model and the polynomial approximation.

This very simple model for the rotation provides a benchmark for the lowest possible performance that a relatively unskilled author of a malicious application could achieve. A more complex process for comparing the templates within the model and those provided by a captured time-series could exploit Dynamic Time Warping (DTW) [39], [40], [41] in order to reduce the effect of any variation in the typing speed between bigrams. DTW is a process by which the similarity between two timeseries can be evaluated independent of non-linear variations in their time dimension. Essentially this approach is used to 'mitigate against distortions in the time axis' [42]. There are several other variants on DTW (e.g. weighted and derivative DTW) that can be used when comparing timeseries. However Lines and Bagnall [43] suggest when there is a warping window size that is set through cross-validation then DTW is 'commonly accepted as the gold standard against which to compare alternative measures.' This allows us to calculate the similarity between the training data and a later sampled timeseries, independent of the time taken to tap the bigrams. In the simple model we assume that any variation in the time taken to type a bigram is inherently linear, i.e. if an individual takes 20% longer to type a bigram this increase in time is distributed evenly across the bigram. Using Dynamic Time Warping, rather than the RMS error, to

calculate the similarity between a measured timeseries and a given bigram from the training data allows us to remove this assumption. This allows us to consider the non-linear variation in the timeseries from the sensors, however there is a significant impact on the computational requirement.

The use of DTW does however significantly increases performance across the four experiments as shown in table 8, indicating there is a significant non-linear component in the time dimension of the sampled data. This shows that if we have the complete word in the training set we can, on average, accurately identify over 81% of bigrams, if we have seen the bigram before but not the word in which it appears we can identify, on average, nearly 30 % of bigrams. It is worth noting that this is the average performance and the variation from the mean is difficult to predict for an individual user. However, there is some evidence that larger devices lead to a greater accuracy as do users who type with their fingers whilst holding the device in one hand.

TABLE 8
The average accuracy using DTW across the different experiments.

Experiment	Accuracy (%)	Performance change (%)
Naive training	64.6 %	30.58 %
Bigram training	81.0 %	16.25 %
Naive unseen	18.9 %	80.29%
Bigram unseen	29.5 %	83.20 %

4.6 Comparative results

As previously discussed in Section 2 there are a number of existing approaches to keystroke inference using smartphone motion sensors. The research presented in this paper offers a number of improvements over existing studies both in terms of the robustness of the approach and also the results achieved.

TouchLogger [21] focused on determining the keystrokes on a smartphone soft keyboard using the accelerometer alone. This method was capable of inferring more than 70% of a keystrokes. However, this approach was limited to only using a numeric keypad, whereas our work uses an alphanumeric keypad. The work carried out in TouchLogger, by Cai and Chen, used a single device to collect data, an HTC Evo 4G. The work presented in this paper used an Android application that was freely available and as such we were able to infer results from a variety of devices and also away from strict experimental conditions; participants were free to hold the device in the way that most comfortable to them.

The work of Xu et al. [22], TapLogger, uses a similar approach to Cai and Chen [21] in that they aim to infer keystrokes on a numeric keypad. However, like our own research Xu et al. make use of both the accelerometer and gyroscope. The experiments that they carried out were in a controlled environment on two specific devices (Google Nexus (One) and an HTC Aria). The training data amounted to approximately 400 taps, significantly more than our approach, which required less than 140 characters. Our own research divides keystrokes in bigrams, whereas Xu et al. aim to extract single taps. This approach has the potential to be affected by linear drift. The experiments carried out as part of TapLogger show accuracies of between 70% and 99%, depending on the user. Whilst our approach may not have the same levels of accuracy (as seen in Table 8), we believe that our research offers a robust and cross-device approach capable of inferring alphanumeric keystrokes.

Perhaps the closest work to our own is that of Shen et al. [24], this approach still looked to infer keystrokes on a numeric keypad but worked with a number of different devices, data sizes and sampling rates. Their work is capable of detecting when a keystroke has occurred 100% of the time, which is comparable with our own results. The accuracy achieved was between 71.4% and 83.9%, but this was dependent on the conditions. Again, the experiments for this work were carried in a controlled environment; while there were three different devices used this is still not comparable to our own open experiment.

5 Implications

The naive approach to protecting against this attack is through requiring an application to explicitly request permission to access the motion sensors for a device. However, it could be argued that this is largely impractical since the use of these sensors is so pervasive within applications and is key to many of the 'seamless' parts of the mobile experience and increasingly leveraged through higher level APIs such as the Android Activity detection.

Many applications will also have valid requirements for creating background processes with listeners attached to the motion sensors, for example fitness trackers. However, it is more realistic to require these applications that launch background process which then attach to sensors to explicitly request permissions. This would reduce the main threat demonstrated in this research (that of the leakage of information from one sensitive application to a background process started by a malicious application).

The granting of permissions for background applications to access a different activity is already performed with the 'accessibility' function, in which an application requests the ability to read from a different 'Activity' — it should be noted, however, that the user experience for this process is generally poor. It is also worth noting, that this would result in a complicated situation where the access to the sensors may or may not require permissions depending on a fairly technical description on how it is accessed — this is not an ideal situation for the common user of these devices who is already struggling to make rational decisions from this permissions model [10].

The current trends are for devices to have more accurate, lower noise and faster sampling sensors — largely driven by the desire for virtual or augmented reality experiences. Schemes such as Google's *Daydream* specification are encouraging devices to be better at sensing both the world around them and its position and movement through the environment. The ability to better sense the world around the device will, inevitably, increase the potential for sensitive information to leak from within an application through this channel.

6 Conclusion

The research presented in this paper has demonstrated how it is possible to estimate the text that has been typed purely from the motion sensors — explicitly the gyroscope and accelerometer. Using a very short training dataset (less than the size of a tweet) we were able to correctly infer, on average, over 81% of bigrams forming words that were part of the training set. We were able to correctly identify nearly 30% of the bigrams in words that were not part of the training data. Whilst this may appear low, it is still of concern given the very small training size and the fact that typically languages have significant redundancy in both word and sentence structure [44], [45], [46], [47], which often allows pragmatics to be extracted from incomplete text.

This attack could potentially lead to the leakage of sensitive data from a secure application through this 'zero-permission' channel. From our experiment we were not able to identify any particular characteristics of users or devices that were more vulnerable than others, however we did find that the accuracy tended to be higher for larger devices and users who used their fingers rather than thumbs to type.

This research highlights the risk of leakage of potentially sensitive information from one application to another, potentially malicious, application. The results have shown that even with very simple models a good level of accuracy can be achieved, by using more computationally expensive approaches such as dynamic time warping this can be significantly improved. It is also worth noting that this research does not leverage the inherent redundancy within written languages which means that there is no requirement for 100% accuracy to infer semantic meaning from the content.

In this paper we also highlight some of the challenges for defending against these attacks, a number of potential approaches have been suggested although most are challenging or would have significant impacts to either the user experience or the Android ecosystem.

Acknowledgments

The authors would like to firstly thank those who took part in this study, without them this research would not be possible. The authors would also like to thank those in the Centre For Electronic Warfare, Information and Cyber for their input on this research, we would also like to thank colleagues in the Digital Forensics group for their input in the initial phases of this research.

We would also like to thank the anonymous reviewers who provided feedback and helped to improve this paper.

References

[1] O. Peters and S. B. Allouch, "Always connected: A longitudinal field study of mobile communication," *Telematics and Informatics*, vol. 22, no. 3, pp. 239 – 256, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736585304000607

[2] A. K. Karlson and S. Brush, B. A. J.and Schechter, "Can I borrow your phone?: Understanding concerns when sharing mobile phones," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 1647–1650. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518953

[3] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner, and K. Beznosov, "Android permissions remystified: A field study on contextual integrity." in *USENIX Security*, vol. 15, 2015.

[4] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev, and C. Glezer, "Google Android: A comprehensive security assessment," *IEEE Security & Privacy*, vol. 8, no. 2, pp. 35–44, March 2010.

[5] Y. Wang, J. Zheng, C. Sun, and S. Mukkamala, *Quantitative Security Risk Assessment of Android Permissions and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 226–241. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-39256-6_15

[6] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 627–638. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046779

[7] W. Enck, M. Ongtang, and P. McDaniel, "Understanding Android security," *IEEE Security & Privacy*, vol. 7, no. 1, pp. 50–57, Jan 2009.

[8] A. Lepp, J. Li, J. E. Barkley, and S. Salehi-Esfahani, "Exploring the relationships between college students cell phone use, personality and leisure," *Computers in Human Behavior*, vol. 43, pp. 210 – 219, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747563214005822

[9] A. Smith, "US smartphone use in 2015," Pew Research Center, 2015.

[10] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, ser. SOUPS '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:14. [Online]. Available: http://doi.acm.org/10.1145/2335356.2335360

[11] R. Yu, "The secrets of malware success on Google Play Store," in *RSA Conference*, March 2016.

[12] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "ECDSA key extraction from mobile devices via nonintrusive physical side channels," Cryptology ePrint Archive, Report 2016/230, 2016, http://eprint.iacr.org/2016/230.

[13] C. Bevan and D. S. Fraser, "Different strokes for different folks? Revealing the physical characteristics of smartphone users from their swipe gestures," *International Journal of Human-Computer Studies*, vol. 88, pp. 51–61, 2016.

[14] O. Miguel-Hurtado, S. V. Stevenage, C. Bevan, and R. Guest, "Predicting sex as a soft-biometrics from device interaction swipe gestures," *Pattern Recognition Letters*, vol. 79, pp. 44–51, 2016.

[15] T. Iso and K. Yamazaki, "Gait analyzer based on a cell phone with a single three-axis accelerometer," in *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM, 2006, pp. 141–144.

[16] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[17] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 397–413.

[18] ——, "The perils of user tracking using zero-permission mobile apps," *IEEE Security Privacy*, vol. 15, no. 2, pp. 32–41, March 2017.

[19] A. Hojjati, A. Adhikari, K. Struckmann, E. Chou, T. N. Tho N., K. Madan, M. S. Winslett, C. A. Gunter, and W. P. King, "Leave your phone at the door: Side channels that reveal factory floor secrets," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 883–894. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978323

[20] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(Sp)iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 551–562. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046771

[21] L. Cai and H. Chen, "Touchlogger: Inferring keystrokes on touch screen from smartphone motion." *HotSec*, vol. 11, pp. 9–9, 2011.

[22] Z. Xu, K. Bai, and S. Zhu, "Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors," in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 2012, pp. 113–124.

[23] A. J. Aviv, B. Sapp, M. Blaze, and J. M. Smith, "Practicality of accelerometer side channels on smartphones," in *Proceedings of the 28th Annual Computer Security Applications Conference*. ACM, 2012, pp. 41–50.

[24] C. Shen, S. Pei, Z. Yang, and X. Guan, "Input extraction via motion-sensor behavior analysis on smartphones," *Computers & Security*, vol. 53, pp. 143–155, 2015.
[25] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang, "Accessory: password inference using accelerometers on smartphones," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. ACM, 2012, p. 9.
[26] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tapprints: your finger taps have fingerprints," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACm, 2012, pp. 323–336.
[27] S. Narain, A. Sanatinia, and G. Noubir, "Single-stroke language-agnostic keylogging using stereo-microphones and domain specific machine learning," in *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks*. ACM, 2014, pp. 201–212.
[28] P. M. Fitts and M. I. Posner, "Human performance." 1967.
[29] A. Dvorak, N. Merric, W. Dealey, and G. Ford, *Typewriting Behaviour*. American Book Company, 1936.
[30] S. Douhou and J. R. Magnus, "The reliability of user authentication through keystroke dynamics," *Statistica Neerlandica*, vol. 63, no. 4, pp. 432–449, 2009. [Online]. Available: http://dx.doi.org/10.1111/j.1467-9574.2009.00434.x
[31] O. Buckley and D. Hodges, "Keystroke inference using smartphone kinematics," in *Human Aspects of Information Security, Privacy, and Trust*, 2017, to Appear.
[32] D. T. Wagner, A. Rice, and A. R. Beresford, "Device analyzer: Understanding smartphone usage," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 2013, pp. 195–208.
[33] L. A. Goodman, "Snowball sampling," *The Annals of Mathematical Statistics*, vol. 32, no. 1, pp. 148–170, 1961. [Online]. Available: http://www.jstor.org/stable/2237615
[34] AppBrain. [Online]. Available: https://www.appbrain.com/stats/top-manufacturers
[35] C. E. Metz, "Basic principles of ROC analysis," *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283 – 298, 1978.
[36] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786550500303X
[37] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982. [Online]. Available: http://dx.doi.org/10.1148/radiology.143.1.7063747
[38] P. Yin and X. Fan, "Estimating r 2 shrinkage in multiple regression: A comparison of different analytical methods," *The Journal of Experimental Education*, vol. 69, no. 2, pp. 203–224, 2001.
[39] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.
[40] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.
[41] E. Keogh, "Exact indexing of dynamic time warping," in *Proceedings of the 28th International Conference on Very Large Data Bases*, ser. VLDB '02. VLDB Endowment, 2002, pp. 406–417. [Online]. Available: http://dl.acm.org/citation.cfm?id=1287369.1287405
[42] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 2005, pp. 506–510.
[43] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 565–592, May 2015. [Online]. Available: https://doi.org/10.1007/s10618-014-0361-2
[44] D. Zola, "Redundancy and word perception during reading," *Perception & Psychophysics*, vol. 36, no. 3, pp. 277–284, 1984. [Online]. Available: http://dx.doi.org/10.3758/BF03206369
[45] A. Kundu and Y. He, "On optimal order in modeling sequence of letters in words of common language as a markov chain," *Pattern Recognition*, vol. 24, no. 7, pp. 603 – 608, 1991. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0031320391900273
[46] M. Taft and K. I. Forster, "Lexical storage and retrieval of prefixed words," *Journal of Verbal Learning and Verbal Behavior*, vol. 14, no. 6, pp. 638 – 647, 1975. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002253717580051X
[47] G. A. Miller and J. A. Selfridge, "Verbal context and the recall of meaningful material," *The American Journal of Psychology*, vol. 63, no. 2, pp. 176–185, 1950. [Online]. Available: http://www.jstor.org/stable/1418920

Duncan Hodges received an M.Eng. degree in electronic and communications engineering (first class honors) and a Ph.D. degree in Electronic and Electrical engineering from the University of Bath, Bath, U.K., in 2002 and 2006, respectively.

He was a Research Officer in the Department of Electronic and Electrical Engineering, University of Bath, before taking a research post working for the UK Government. Following this post he returned to academia as a Senior Researcher at the Cyber Security Centre at the University of Oxford. He is now a Lecturer in the Information Operations Group within the Centre for Electronic Warfare, Information and Cyber at Cranfield University. He is currently based at the Defence Academy of the UK.

His current research interests include cyber security as a socio-technical problem, the application of machine learning and natural language processing within this domain as well as notions of identity in on- and off-line spaces. He has an ESRC NCRM Fellowship and is a visitor at the Turing Institute.

Oliver Buckley holds a BSc Computer Science (first class honours) from the University of Liverpool in 2004 and a Ph.D. degree in Computer Science from the University of Wales, Bangor, U.K. in 2009, specialising in soft tissue deformation simulation using haptics and visualisation.

He has experience within industry as a software engineer and in academia. This has included working on the Corporate Insider Threat Detection project as a Cyber Security Researcher with the University of Oxford, Oxford, U.K. He is currently a Lecturer the University of East Anglia in the School of Computing Sciences. Prior to this he worked at Cranfield University, where he was a member of the Information Operations research group.

Oliver's current research interests include cyber security, behavioural biometrics, with a current focus on keystroke dynamics, as well as the application of machine learning and visualisation within the domain.