In Depth Review of “Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches”

Wesley Lam

Abstract—The following paper is an indepth review of *Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches* written by Anindya Maiti, Murtuza Jadliwala, Jibo He, and Igor Bilogrevic. These individuals are referred to as “the authors” in this paper, and *Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches* is referred to as the *original paper* in this review. The original paper covers the use of side channel attacks on smartwatches to gain information on smartphone handheld numeric touchpad key presses. This review cover each section of the original paper chronologically by section, first with a summarization of that section, second with following comments that will either criticize, praise, give a suggestion for improvement, or divulge deeper into a topic that may or may not have been discussed in the original paper. Then an additional remarks section is included which comments on the entire original paper as a whole.

1. Introduction

The introduction brings awareness of the popularity of smartwatches as well as the variety of sensors available in smartwatches. The availability of GPS, microphones, cameras, accelerometers, and gyroscopes are common among smartwatches. However, gyroscopes and accelerometers are to sensors that cannot be controlled or disabled by the users [1]. This introduction provides a convincing argument of performing side-channel attacks on smartwatches, as there is a large potential target pool, as well as a higher likelihood of a successful attack since the user cannot easily defend themselves as they cannot access gyroscope and accelerometer permissions. Before divulging into their own work, the authors wisely present similar past projects other groups have done which used only smartphone motion data instead of smartwatches motion data. This allows a baseline to compare their own results to other research in side channel attacks on mobile keypads.

2. Related Work

The beginning of the related works section lists a variety of side-channel attacks research in the past on different devices such as printers or LED monitors. Afterwards the authors state the different potential side-channel attacks on smartphones and their damages such as GPS with location privacy, microphone recordings and valuable ambient sound information, and smartphone cameras and inference to keystrokes.

Many different inference attacks on obtaining keystroke information such as using acoustic signals or electromagnetic emanations from external keyboards from previous works are introduced as well. While the authors and the other similar works have the same goal of obtaining keystroke information, the authors distinguish their inference attack methods from others by using smartwatch motion sensors data as their source of information. The related works section provides a more in-depth knowledge to previous projects that use side-channel attacks with specific examples like extracting optical information from monitors or eyes to infer information about connect being watched. This knowledge provides a good background into side channel attacks. It also allows the authors to distinguish their side-channel attack method from others to make it clear that the project they are researching is not a replica of a prior project in the same area.

3. Attack Description

The third section of *Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches* describes the specific scenarios that are analyzed to perform the smartwatch side-channel attack. The three scenarios are based on different hand configurations, which are categorized as Smartphone and smartphone on:

a) Same Hand and Non-Holding Hand Typing (SH-NHHT)
b) Same Hand and Holding Hand Typing (SH-HHT)
c) Different Hand and Non-Holding Hand Typing (DH-NHHT)

These different hand configurations can be observed from figure 1, which is from the original paper:

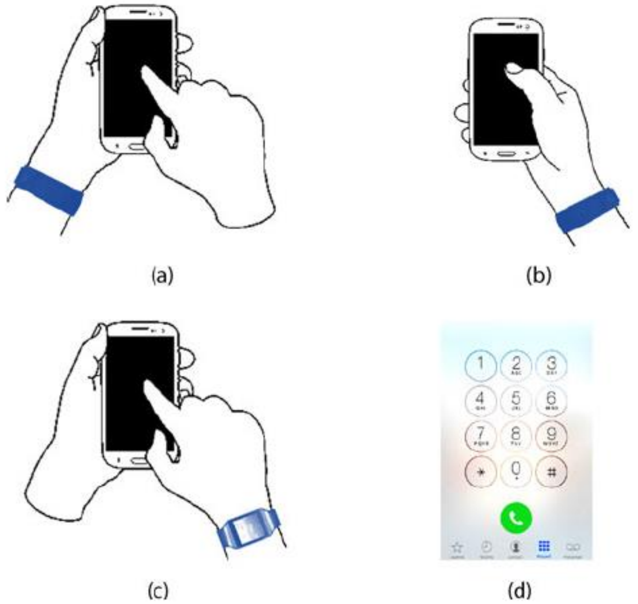


Fig. 1. Smartwatch and smartphone on (a) Same Hand and Non-Holding Hand Typing (SH-NHHT), (b) Same Hand and Holding Hand Typing (SH-HHT), (c) Different Hand and Non-Holding Hand Typing (DH-NHHT), and (d) numeric keypad used in our experiments.

32.83% of people use hand configuration a, with 16.17% of people using the opposite or vice versa position of a (figure 2).

28.44% of people use hand configuration b, with 7.56% of people using the opposite or vice versa position of b (figure 3).

2.11% of people use hand configuration C.

However, configuration C is not analyzed the same way as configuration A and B since key press events cannot be accurately detected if the experiment is set up the same way as configuration A and B.

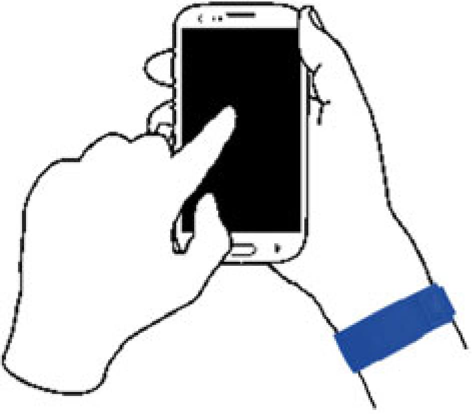


Fig 2. Variation of typing scenario in Fig 1a. (Vice Versa scenario)
From original paper, as Fig 13 [1]

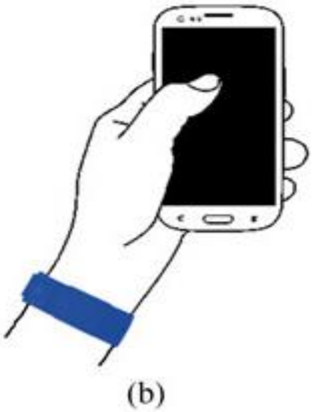


Fig 3. Variation of typing scenario in 1b. (Vice Versa scenario)
From original paper, as Fig 13 [1]

With this information 44.61% of left handed smartwatch users and 40.39% of right handed smartwatch users are potential targets for the side-channel attack. This means overall 85% of smartwatch users are targetable for the side-channel attack. To further improve odds of performing a successful side-channel attack on any user, more hand scenarios should be considered and analyzed. Otherwise in the case an adversary wanted to perform a side channel attack on an individual whose hand configuration did not match any prior hand configuration mentioned previously, then the adversary could not perform the attack with meaningful results. It is understandable the authors did not pursue further hand configurations for the sake of time, as 85% of the hand configuration population is already a significant amount. However if further efforts to continue this project exist, then one step to produce more significant results is to spend more time in studying more hand configurations to raise the potential population percentage above 85%.

A large presumption is that the authors assume that the target has installed a malicious app or trojan software to gain access to gyroscope and accelerometer sensors. While this situation is certainly feasible, the entire side-channel attack depends whether or not this presumption is successful. The information from the malicious app needs to be sent to the attacker, which means it likely has to be a malicious app that the attacker developed, or allows the attacker to somehow have access to the gyroscope and accelerometer readings. The authors can enhance the credibility of their research by developing an example malicious app to be used. Instead they simply developed an app that records linear accelerometer measurements. Getting users to install the software itself is another issue itself. Users need to somehow be introduced to the app one way or another, in addition the user needs to be convinced to download and install the app. Here, there is already an issue that makes the attack attempt difficult. So far the following preconditions exist for a successful attack:

1. The adversary must know the users hand configuration beforehand, and the user must use one of the hand configurations studied (85% chance for attack)
2. The adversary must somehow access the user's smartwatch accelerometer and gyroscope data, which is done with malicious software
3. To do this, the adversary must first find a way to present the software to the user
4. The user must then decide to install the software or the adversary can forcibly install the software onto the users smartwatch covertly which may be difficult

These preconditions make it difficult for the adversary to perform a large-scale attack with the data of many users being compromised at once. There are a few potential scenarios where this can occur however, such as sending out a large-scale scam email that advertises the malicious app or advertising the app itself for users to download. However whether or not the adversary has the financial resources to perform the latter is another different matter. To obtain the large scale hand configuration data, a survey could be a prerequisite to fill out on the users mobile or smartwatch device to download the app, or the app could require the user to select or somehow indicate how they hold their phone (if applicable since it would be a smartwatch app) and how they wear their smartwatch. Of course even in this scenario, the user could simply input false information into the survey which would provide the adversary with invalid data.

If the adversary cannot meet the conditions of presenting the malicious software on a large scale and obtaining hand configuration data of individuals on a large scale, then the chance of performing a large scale side-channel attack is highly unlikely. Therefore it would be expected that this kind of attack would be performed on specifically targeted individuals. Additionally even if a successful attack occurs, the user still has the possibility of noticing the information transfer between the app and the adversary if a covert channel is not used [1], which may stop the attack indefinitely if the user uninstalls the app. Finally even if a successful attack occurs, to obtain meaningful numeric keypad information, the user must actually use the numeric keypad. Newer smartphones have fingerprint identification, which allows the user to skip pin authentication which traditionally used a numeric keypad. It is probable that not all victims use their phones numeric keypad to insert information like credit card numbers or other sensitive information. As most services that require the use of numeric keypad inputs could be done through other means. For example consider the scenario of a user purchasing something over the phone, they may use the numeric keypad to fill in their credit card information. However that user could also never make purchases via keypad presses and simply make purchases online with a virtual keyboard on their phone.

4. Classification-Based Attack Framework

Section 4 of *Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches* covers how the data is recorded and processed, as well as how the classifiers are trained and combined. The three dimensional acceleration is measured through a smartwatch app and the keypad information is measured through a smartphone app that is installed on the smartphone paired with the users smartwatch. Two algorithms are discussed to identify how a “keystroke-record” is recorded. The first algorithm examines the energy or sum of acceleration on the three axis and algorithm 2 determines a threshold value of deciding how much energy is needed for an actual keystroke. If the energy value surpasses this threshold, then that acceleration data is saved since it corresponds to a keystroke. Through observations, a 50 Hz sampling frequency is found to provide sufficient accuracy in capturing related keystroke data. To prevent accidentally recording the same keystroke too many times, after a keystroke is captured the next 14 samples of data are ignored. In the preliminary work 54 basic time domain features are used to identify unique key presses such as maximum, minimum, mean, median and etc. of magnitude samples. These different time domain classifications are referred to as feature vectors. The feature vectors are labeled and used to train classifiers. The authors develop five different classification algorithms:

1. Simple-Linear Regression (SLR)
2. Random Forest (RF)
3. K-nearest neighbors (k-NN)
4. Support vector machine (SVM)
5. Bagged Decision Trees(BDT)

Due to the strengths and weaknesses of each classifier, an ensemble classification approach is used instead which takes advantage of all five classifier algorithms. The five different classifiers are trained individually then combined together in a *majority wins* ensemble classification, similar to a voting mechanism where the majority is the winner.

The five different classifiers undergo supervised machine learning to train themselves to identify unique key presses more accurately. The combination of the 54 time domain features along with the ensemble classification approach provides an extremely impressive method for identifying key presses. Not only does it enable the user of every classifier, it uses and combines the results of every classifier to produce an even stronger classifier known as the ensemble classifier. The only potential threat of an ensemble classifier is if the majority of its source classifiers produce incorrect results. This would lead to an overall worse classifier since the results of the ensemble scheme would overshadow the results of the source classifier that is actually producing correct predictions. However this can be easily avoided if the results of each individual classifier are compared with the ensemble classifier which was done in the original paper. Fortunately the results were more robust with the ensemble method, so the overall prediction rate can be increased. Otherwise only one or a few classifiers could be used as the others would need to be thrown out as they would only provide weaker results.

5. Evaluation of Classification-Based Attacks

The section 5.1 or the experimental setup section discusses the participant demographic, the specific equipment used, the software used, the hand configuration, and the sampling frequency used with certain alterations for the experiments in section 5.4, 5.6, 5.7 and 5.8. For the section experiments (except the alteration sections), the participant demographic consisted of 12 participants aged 19-32. Hand formations of SH-NHHT (fig 1a) and SH-HHT (fig 1b) were tested. A Samsung Gear live smartwatch was used equipped with an InvenSense MP92M 9-axis Gyro + Accelerometer + Compass sensor. The Motorola XT1028 smartphone was used for the numeric keypad. The linear accelerometer data from the smartwatch was sampled at 50Hz. Weka 3.7.2 libraries were used to train and test classifiers. Finally MATLAB R2014a was mostly used to compute time and frequency domain features.

Section 5.2 discusses the different training datasets used to construct the classifiers as well as the testing procedure. Participants typed in uniformly distributed random numbers between 0-9 from an audio stream. The different training/testing scenarios are listed as *One versus One*, *One versus Rest*, and *All Versus All*.

One versus One trains the classifier with the data from one single target (participant in this case), and then uses that data to infer the key presses from that target.The *One versus One* is the best case scenario since the training data originated from the very same target the classifier is trying to infer key presses from. *One versus One* classification accuracy on average was 84.58% for SH-NHHT and 83.5% for SH-HHT. One issue of *One versus One* is that it would be difficult to obtain the training data set from the very target an adversary is after. This implies that the adversar*y already knows* what the target is key pressing, and with this known data, the adversary would then train the classifiers with it. The setup of *One versus One* defeats the purpose of an adversary trying to identify the key presses of a target since they need to already know the key presses of a target in order for a *One versus One* training scenario to occur. In other words, the process of setting up a *One versus One* scenario includes knowing a target's key presses already, which is the entire goal of the side-channel attack in the first place. Thus, a *One versus One* training scenario is extremely unlikely. However it is not impossible for this situation to occur. For example in a situation where an adversary is able to use video surveillance of a target with their smartphone screen in frame, and the target already has the malicious software installed, the adversary could train the classifier in a *One versus One* scenario by matching the timing of the key presses in the video surveillance to the input data from the linear accelerometer. This could be useful because the time of when a key press is done could reveal different information. For example obtaining video surveillance of key presses during work hours may only reveal a target's pin information to unlock the smartphone. Then if the adversary uses the *One versus One* trained classifiers and performs a side-channel attack during the night, it may be possible to obtain different key press information more accurately like a credit card number. Likewise in any other situation the adversary can obtain any accurate key press information with linear accelerometer information beforehand, a *One versus One* trained classifier side-attack could be used to infer <u>future</u> key presses more accurately.

One versus Rest trains the classifier with the data from other participants not including the target. The *One versus Rest* training scenario is a much more realistic training set to be used since the first time the adversary approaches a target, the adversary may not have training data on the target already. On average the *One versus Rest* classification accuracy was 70.08% for SH-NHHT and 71.16% for SH-HHT. The *One versus Rest* case should be highlighted as the primary method of training since it makes more sense for an adversary to go after a target with key press information that they do not already know. Already knowing a target's previous key press information is a much more difficult requirement to meet as opposed to using previously trained data from a large sample against the target. This training scenario is especially true if an adversary is going after a random target. The hand configuration could simply be obtained through observation. However a record of known key presses matched with linear accelerometer data would be significantly harder to obtain without a very detailed attack process in mind.

All versus All trains the classifier with the data from all participants against all participants. In other words, the overall average results from conducting a *One versus All* on every participant would be equivalent to an *All versus All* scenario. A *One versus All* scenario is not covered in the original paper since it would be a redundant scenario to test a classifier with. This is because the conditions for fulfilling a *One versus All* scenario means that one can also satisfy the conditions to create a *One versus One* scenario, which is by far more accurate since it is the best case scenario. On average the *All versus All* classification accuracy was 88.16% for SH-NHHT and 85.83% for SH-HHT. The *All versus All* training scenario is useful if you can obtain a large group of key press data on many targets. This scenario would be used specifically if there is no specific target in mind, with the attacks occurring indiscriminately. Or in a scenario where it is known the target data exists somewhere within the large group of key press data, but it is not known exactly which data corresponds to which person. One issue of this training scenario is that it is not guaranteed that the hand configurations for everyone is already known. If the adversary is planning to attack indiscriminately with *All versus All* trained classifiers, the adversary should know the hand configuration to avoid loss of accuracy in the inference results. Attacking multiple non-specific targets at once implies there is missing information in one regard or another against those targets. Otherwise why would the adversary use the *All versus All* scenario instead of performing multiple *One versus One* scenarios? With the only exception of time, there are almost no logical explanations of using multiple *One versus One* scenarios to obtain greater inference accuracy.

Section 5.3 of the original paper covers the same experiment in section 5.1 and 5.2 but with lower sampling frequency. The reduced frequency experiment drops the sampling rate from 50Hz to 25Hz and 10Hz. Accuracy notably dropped with about 60-80% accuracy for all scenarios at 25Hz, and 10-20% accuracy for 10Hz. With these results, the low sampling rate of 10Hz is not a viable sampling rate. At 25Hz the classification accuracy may not be significant enough depending on the specific training scenario and hand configuration used.

Section 5.4 compares the inference attacks with smartwatches with the inference attacks using smartphones. The method of attack is nearly the same, the only difference is one attack gathers the sensor data from the smartphone, the other gathers the sensor data from the smartwatch. Results appeared nearly the same between smartwatches and smartphones, with only marginal differences. One notable feature of performing the experiment with the smartphone is that some numbers had much greater classification accuracy than others. This feature was not observed in the smartwatch experiment. The results help validate the threat of the authors use of a smartwatch side-channel attack. The smartphone experiments were previously known validated research. This baseline of previous research allows the authors to compare their research to the latest research on key press inference attacks. Since their results are on par with recent research, it is evident that they are not performing outdated or nonsignificant research.

Section 5.5 details the effect of combining smartphone and smartwatch data. The results revealed improved accuracy in comparison against the smartphone-only and smartwatch-only experiments. The improved accuracy was typically marginal, but regardless the results were stronger than before. The authors briefly mention that combined smartwatch and smartphone attacks are realistic since smartwatch operating systems and applications typically link to a smartphone. One challenge they do not mention however, is creating the malicious software that can spread from smartwatch to smartphone. The malicious software can compromise the sensor data of the smartwatch, but for the same malicious software to spread and compromise the smartphone sensor data is a different matter. If the original paper can present existing malicious software that can spread between smartwatches and smartphones, the strength of the paper would be improved. The authors appear to take for granted that spreading and obtaining information between smartwatches and smartphones can be done easily without showing any specific method how.

Section 5.6 discusses the effects of a more natural typing scenario in contrast to the controlled typing scenario in the experiment. In the original experiment (section 5.1 and 5.2) a stream of numbers were relayed via audio in uniform time distributions. The natural typing experiment had participants type at their normal typing pace and overall the results weakened. The overall mean classification accuracy for SH-NHHT was 52% and for SH-HHT was 61%. Note that the original overall classification accuracy for SH-NHHT was 80.97% and for SH-HHT was 80.16%. This provides a concerning drop of about 30% and 20% accuracy for SH-NHHT and SH-HHT respectively.

Section 5.7 performs the same base experiment in section 5.1 and 5.2 but with different adversary and target smartwatches. In the original experiments the adversary and the target used the same smartwatch, but for this case the adversary and the target are strictly using different smartwatches in terms of make and model. The smartwatch used for the targets was the LG Urbane W150 smartwatch with an InvenSense M651 accelerometer and gyroscope equipped while the smartwatch used for the training data was the Samsung Gear Live which was the original smartwatch used in section 5.1 and 5.2. The results of this experiment are shown in figure 4, which was originally figure 6 from the original paper.

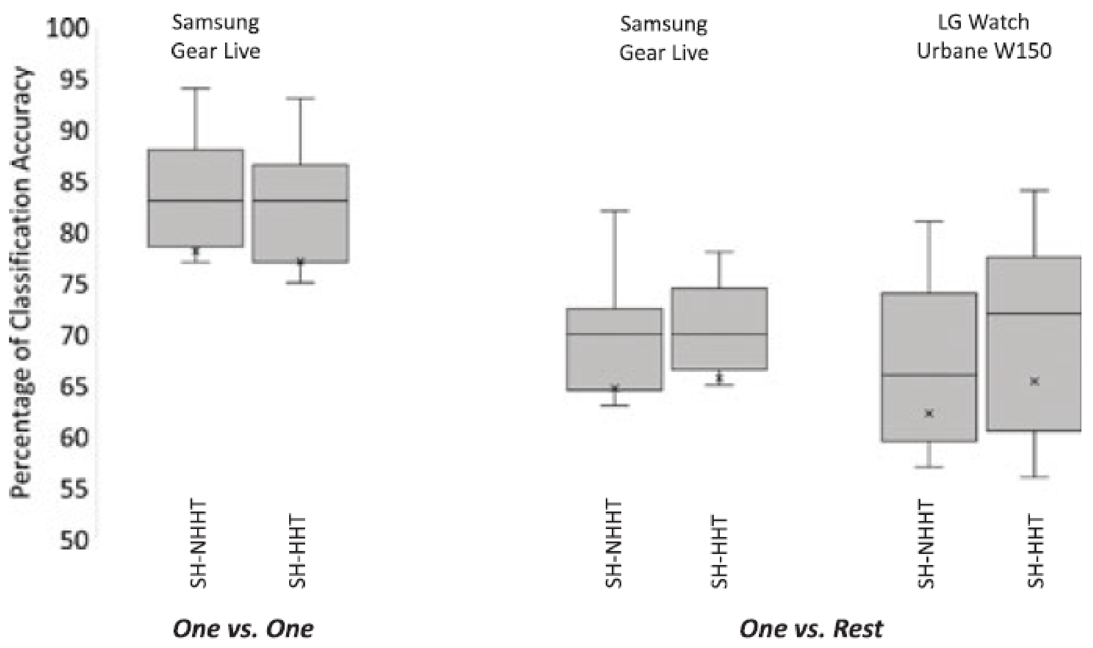


Fig. 4 . Classification accuracy for *One versus One* and *One versus Rest* using two different smartwatches (Samsung Gear Live and LG Watch Urbane W150). (Originally Figure 6 from *Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches* [1])

The mean classification accuracy slightly dropped for the LG Watch Urbane W150 from the original watch, the Samsung Gear Live. The accuracy difference for *One versus Rest* in SH-NHHT was 70.08% versus 67.41% and in SH-HHT 71.16% versus 70.83% for the Samsung Gear Live and the LG Watch Urbane W150 respectively. While the accuracy is very close to each other, note on the variance on the LG Watch Urbane W150 is much larger. With these findings the authors conclude that the keystroke inference with different smartwatches is still feasible. This conclusion is not completely fair as this experiment only compared the different make and model of one other smartwatch. The sample of different smartwatches used is insufficient and unrealistic. Potential targets in this attack are not limited to simply two smartwatches (The Samsung Gear Live and the LG Watch Urbane W150). There exists at least dozens of different smartwatch models as Wikipedia lists over 80 different models of smartwatches to date [2]. Another problem with this incomplete experiment is that the classifiers were trained only on the Samsung Gear Live. The experiment would provide more significant results if the vice versa case was considered where the classifiers were trained using the LG Watch Urbane W150 and then the target data was inferred from the Samsung Gear Live. The linear accelerometer and other sensor data on the

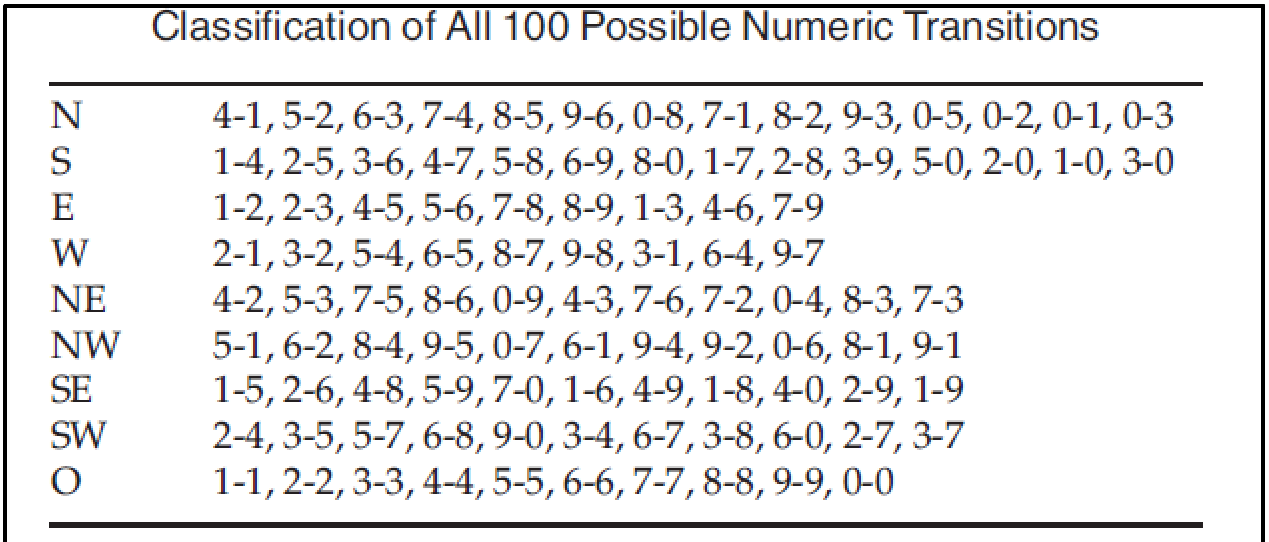
Samsung Gear Live may be more sufficient or easier to obtain in comparison to other smartwatches but with no control let alone experiment to test this, concluding that using smartwatches of different make and models between targets and adversary is weak. Even if the average accuracy among many different smartwatches is almost the same, there may be cases of unacceptable variance between different smartwatch models. Also the authors could make the resulting data more transparent by revealing the *One versus One* and *All versus All* classification accuracy in addition to only the *One versus Rest* classification accuracy for the LG Watch Urbane W150.

Section 5.8 quickly mentions the extension of the experiment to QWERTY keypads,that is instead of inferring numeric keypads, the experiment would try to infer characters on alphanumeric mobile keypads instead. The average classification accuracy was 30.44% which is not significant enough to be used viably in an attack. The authors do mention however potential improvement in accuracy by performing a dictionary-based search and/or analyzing keyboard characteristics.

Section 5.9 simply addresses the vice versa cases of SH-NHHT and SH-HHT (figure 2 and 3) do not hinder the experiment results. A two-tailed t-test is used to prove and make this claim.

6. Relative Transitions-Based Attack Framework

This section addresses the third hand configuration in figure 1c. There exist typo mistakes in the beginning of section 6 as the original paper mentions a DH-HHT or Different Hand and Holding Hand Typing configuration which is not a configuration mentioned in the paper. They refer the DH-HHT to figure 1c which is the DH-NHHT configuration not the DH-HHT configuration. Since the DH-NHHT configuration cannot detect key press events with the same setup as the SH-NHHT and SH-HHT configuration. The SH-NHHT and SH-HHT configuration receive their results based on wrist movement. The DH-NHHT configuration instead depends on transitional movement between key presses. Therefore the preprocessing of the DH-NHHT configuration is different as it depends on data from two consecutive key presses as opposed to one like in the SH-NHHT and SH-HHT configurations. Cardinal directions and O (repeat) were used to identify possible transitions. A list of all possible transitions and their classifications are seen in figure 5. Tracing is performed to eliminate key presses of the same classification, for example pressing 1 twice and 2 twice contain the same transition so there must be a way to distinguish between which number was pressed twice. The tracing attempts to predict the correct transition by using data from the previous and following transitions. Three different tracing mechanisms were created:



Classification of All 100 Possible Numeric Transitions

N	4-1, 5-2, 6-3, 7-4, 8-5, 9-6, 0-8, 7-1, 8-2, 9-3, 0-5, 0-2, 0-1, 0-3
S	1-4, 2-5, 3-6, 4-7, 5-8, 6-9, 8-0, 1-7, 2-8, 3-9, 5-0, 2-0, 1-0, 3-0
E	1-2, 2-3, 4-5, 5-6, 7-8, 8-9, 1-3, 4-6, 7-9
W	2-1, 3-2, 5-4, 6-5, 8-7, 9-8, 3-1, 6-4, 9-7
NE	4-2, 5-3, 7-5, 8-6, 0-9, 4-3, 7-6, 7-2, 0-4, 8-3, 7-3
NW	5-1, 6-2, 8-4, 9-5, 0-7, 6-1, 9-4, 9-2, 0-6, 8-1, 9-1
SE	1-5, 2-6, 4-8, 5-9, 7-0, 1-6, 4-9, 1-8, 4-0, 2-9, 1-9
SW	2-4, 3-5, 5-7, 6-8, 9-0, 3-4, 6-7, 3-8, 6-0, 2-7, 3-7
O	1-1, 2-2, 3-3, 4-4, 5-5, 6-6, 7-7, 8-8, 9-9, 0-0

Fig. 5. Classification of all possible numeric transitions. This figure is from the original paper as Table 4 [1].

1. Forward Tracing – transition data is plotted in the same order as they occurred in time
2. Backward Tracing – transition data is plotted in reverse order as they occurred in time
3. Bidirectional Tracing – both forward and backward tracing methods are applied

Bidirectional tracing is used in the evaluations since it is the most accurate and limits transition misclassification errors. The use of relative transitional movement is a smart method of attempting to identify DH-NHHT key presses which could potentially be applied to other hand configurations not studied yet. This brings the challenge of with new hand configurations to consider, new preprocessing methods may need to be discovered.

7. Evaluation of Relative Transition Based Attack

Section 7.1 quickly mentions the experiment setup for the DH-NHHT configuration is the same as the setup in section 5.1 for DH-NHHT with the difference that the smartwatch is worn on the right hand by participants the smartphone held in the left hand.

Section 7.2 discusses the training and testing procedure for the transition based attacks. Numbers are read uniformly in time and randomly like in section 5.2. Out of the total sample of 1200 total numbers typed from the participants, 960 or 80% of the sample data was used for training the transition classifiers. The rest of the data or 20% was used to test the transition classifiers. Key presses are inferred with the requirement that at least the previous or following transition should be identified. The transition classifiers had an overall transition accuracy of 88.42%. There was a noticeable behavior that if one transition was incorrectly predicted, then the next transition would more likely to be predicted incorrectly. This is logical since the outcome is based on previous and future transition predictions. The actual key press inference accuracy was 43.75% (bidirectional tracing). The authors fault the low accuracy due to three primary reasons:

1. Incorrect classified transitions means 2 key presses are wrong, as opposed to only 1
2. Unclassified transitions present missing information
3. A small number of same transition key pairs make it impossible to determine the exact key press

The authors assume that while the key press inference accuracy is low, with multiple attempts the key press information could still be extracted successfully. The real life world examples used was typing in a pin number, password, or card number incorrectly due to human error, causing the need to type in a repeated sequence of numbers. This assumption would allow the adversary to eventually converge to specific correct key press inputs, only if the target uses the same key press sequence multiple times. The weakness of course is that, if the target only uses a certain sequence of key presses once or only a few times with the DH-NHHT configuration, then the adversary would most likely not be able to extract that key press sequence. With such a low inference accuracy of 43.75%, identifying the key presses of DH-NHHT configuration targets appears unreliable with the exception of a prolonged side-channel attack.

Section 7.3 explains how smartwatch and smartphone data was combined for the DH-NHHT scenario. Smartphones cannot capture transitional wrist movement so the data processing is different than the data processing in the SH-NHHT and SH-HHT scenarios, which merged feature vectors from the linear accelerometer. Instead, the combined smartwatch and smartphone data processing works together using a two-step system. First the smartwatch presumes a key pair based on transitional movement, second, if there are contending key-pairs (key pairs that share the same transitional movement) or undetected key pair transitions, then the keystroke record of the smartwatch is referred to fill in the missing or incomplete data. The results of combing smartwatch and smartphone for the DH-NHHT hand configuration is a 82.50% accuracy, much greater than the previous 43.75% accuracy using only smartwatch transitional data. One concern with these results is that it is not clear how accurate smartwatches perform for the DH-NHHT hand configuration alone. Previously in the paper it is very transparent how the smartwatch only classification performs for the SH-NHHT and SH-HHT hand configurations. This can be seen in table 2 in the original paper which compares the accuracy of using a smartwatch only, a smartphone only, and combining the two devices. The table is reproduced in this paper as table 1 for convenience.

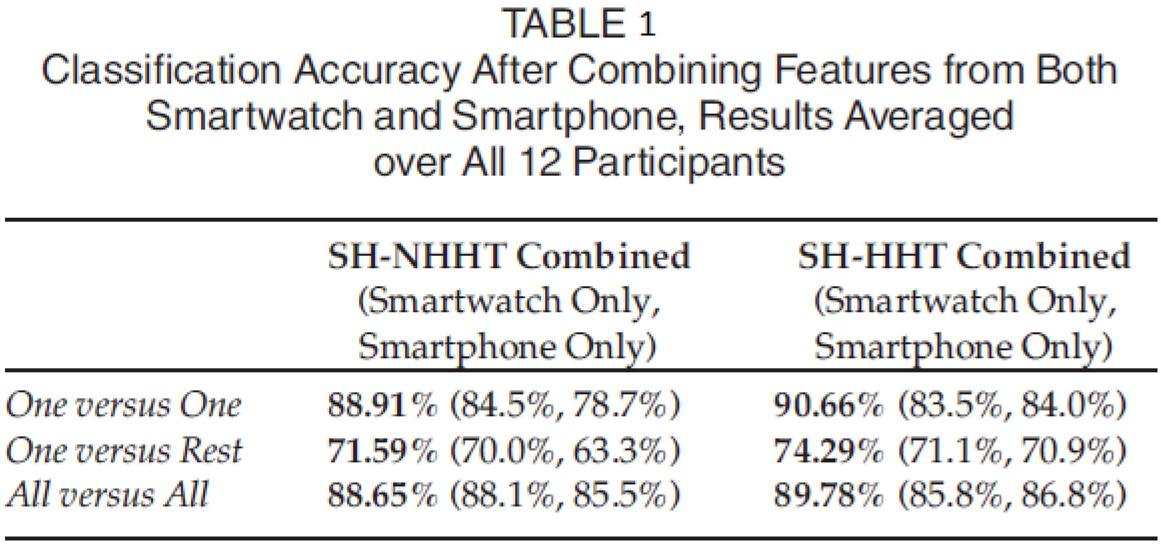


TABLE 1
Classification Accuracy After Combining Features from Both Smartwatch and Smartphone, Results Averaged over All 12 Participants

	SH-NHHT Combined (Smartwatch Only, Smartphone Only)	SH-HHT Combined (Smartwatch Only, Smartphone Only)
One versus One	**88.91%** (84.5%, 78.7%)	**90.66%** (83.5%, 84.0%)
One versus Rest	**71.59%** (70.0%, 63.3%)	**74.29%** (71.1%, 70.9%)
All versus All	**88.65%** (88.1%, 85.5%)	**89.78%** (85.8%, 86.8%)

There are no efforts to discuss how the smartphone performs in the DH-NHHT scenario alone, which may imply that the combined smartwatch and smartphone set-up for the DH-NHHT scenario may heavily rely on the results of the smartphone. For example, if the accuracy of the smartphone alone for DH-NHHT was 81%, and the new accuracy results for using both the smartphone and smartwatch is 82.50%, then the significance behind the transitional classification for the smartwatch is extremely lowered. There is no clear reference to compare the results of combing the smartwatch and the smartphone in this scenario with exception to the smartwatch only scenario, which may be skewed depending on the smartphone only results. Regardless, the results for the combined smartphone and smartwatch experiment prove that the DH-NHHT hand configuration scenario can be analyzed much more reliably.

In section 7.4 the same scenario as 7.2 is considered except with non-restricted or natural typing speed. The transitional classification accuracy dropped from 88.42% to 79.76% and the key press inference classification accuracy dropped from 43.75% to 38.33%. Once again the results support the difficulty in inferring the key presses for DH-NHHT users. Without the use of supporting data from a smartphone, as discussed in section 7.3, DH-NHHT key press would be difficult to infer without a very large sample of data from the smartwatch.

8. Discussion

Section 8.1 reflects on the limitations of the entire project as a whole. The authors consider the situation where the target has adjusted in body posture and orientation. The different body postures and orientations result in different movement patterns, which results in the classifiers mislabeled the incoming data. To counter this multiple classification models need to be created depending on the posture of the victim like (walking, running, sitting in a vehicle, etc). Other similar frameworks from prior literature suffer the same issue. The need for additional classifiers adds many layers of complexity, as each person could behave differently depending on the posture. This section also addresses the issue of power consumption. The battery inside smartwatch deplete very quickly, such as the 31% depletion/hour for the Samsung Gear Live used in the experiments, which occurs when readings are taken at 50Hz. This rapid depletion of battery may alert the target that they are being attacked. To avoid this, a smaller sampling rate or a smart mechanism to record only when the target is typing is necessary. The authors also address that they have not addressed the both hands typing hand scenario, but they wish to work on this in the future. Finally the authors address threats to validity since most of the experiments performed do not address realistic scenarios. They mention some realistic scenarios were observed, such as the natural typing experiments.

The authors do address multiple limitations to their work but one major issue is that they only address each threatening scenario one case at a time. The situation where multiple of these threatening scenarios occur at once is very realistic and not accounted for. Each threatening realistic scenario that reduces the accuracy of the ideal controlled experiment setup can easily cascade onto each other resulting in an enormous accuracy loss. For example, a situation where the target:

1. Changes posture and orientation.
2. Types at varying speeds, per se fast to slow, then slow to fast for example.
3. Varies their hand configuration while the side channel attack is performed.

This situation where a target performs all three of these actions in one observation is realistic and could occur very reasonably. The only method to obtain accurate information is to obtain a method to observe the target or identify how they are behaving at a certain time. The concept of multiple threatening scenarios could provide devastating results and the main issue is that these combined scenarios are completely realistic in nature. Sampling at 50hZ is also an issue, as it would most likely alert a target if their battery levels started to deplete extremely faster without any apparent particular reason. With the current experiment, sampling at the tested 25hZ is more realistic until a smarter mechanism to collect smartwatch linear accelerometer data is implemented. Therefore even without the other scenarios included, the starting expected accuracy should be based on the 25hZ results which already lay in the 60-76% range. Considering just one scenario of using a 25hZ frequency sampling rate, smartwatch data only, a 1 v Rest classifier, and assuming the user is typing at a faster natural speed, the expected accuracy would calculate to (~76% for 25hZ w/ 1 v Rest, 52% natural typing speed for 50hZ, therefore .76*.52 = .3952 or 39.52%) less than 39.52% inference accuracy. Note how it is strictly less than 39.52% since the natural speed typing accuracy was assumed for a 50hZ sampling rate. Other similar experiments set up the same way would yield similar levels of lower inference results. A natural typing experiment combined with lower frequency sampling rates was not performed so the exact inference accuracy for this case cannot be found. However at a 25hZ sampling rate, the natural speed typing scenario inference accuracy will most definitely be lower. The original paper can be improved by performing combined experiments to portray a more accurate representation of realistic scenarios. There are many combined experiments that could potentially be tested. Specifically these combined experiments are not limited to but may include:

1. Smartwatch Only, 25hZ sampling rate, Natural Typing Speed
2. Smartwatch and Smartphone, 25hZ sampling rate, Natural Typing Speed
3. Smartwatch Only, 25hZ sampling rate, Posture/Orientation Change
4. Smartwatch and Smartphone, 50hZ sampling rate, Posture/Orientation Change, Natural Typing Speed
5. Etc.

Section 8.2 quickly mentions a few defense mechanisms against the accelerometer and gyroscope side channel attacks. One proposed defense mechanism is to have the gyroscope/accelerometer with user-defined access controls. Reducing applications sensor sampling rate is another proposed method. A system level monitoring mechanism that can record the sensor data, and potentially flag unwanted access is another proposed method of dense.

Section 8.3 presents a potential stronger tracing method called *Random Walk Tracing* which takes in random subsequences of varying length and performing bidirectional tracing multiple times. The random subsequences cover the entire sequence multiple times. Many samples of random subsequences would be taken, and then any transition with multiple possible key presses could be determined by a majority vote system. This potential upgrade from just bidirectional tracing could increase the accuracy of the present 43.75% inference accuracy of DH-NHHT hand configurations which would be a great improvement. From the hand configurations studied so far, DH-NHHT side channel attacks suffer in reliability due to the low inference accuracy the current attack method yields.

9. Conclusion

Section 9, or the conclusion is self-explanatory as it summarizes the paper in its entirety. This includes the different experiment types like varying hardware, combining smartphone and smartwatch data, and the QWERTY alphanumeric keypad layout.

10. Additional Remarks

One topic not addressed in the original paper is using this type of side channel attack for military or police purposes. Typically performing a side-channel attack like this would assume the attacker is attempting to perform criminal activity like accessing a targets credit card numbers or identify a targets passcode to access their smartphone. However, the use of this type of side-channel attack could potentially be used for good intentions like by the police or military. Gaining information against criminals is a valid way of using these methods of attacks lawfully. Smartphones have the capability of erasing all data stored on themselves if the user inputs certain number of incorrect passcodes. This can provide a backup method for criminals to erase their tracks of illegal activity stored on their smartphone in case their smartphone gets compromised. This exact situation occurred in the 2015 San Bernardino Attack, in which 14 people were killed and 22 were injured due to a terrorist attack. The FBI was able to obtain the attacker’s smartphone, an Apple iPhone 5C, which was suspected to have important information related to the terrorist attack. However, the FBI could not initially unlock the smartphone since they did not know the 4 digit passcode. Additionally the phone was programmed to automatically erase all of its data after ten failed password attempts. This led to complications with a FBI vs Apple court case since Apple did not want to create an updated operating system that could disable security features which was referred to by Apple as “GovtOS.” This is an example of an unpleasant situation that could be avoided if the passcode information was made available earlier which is possible using the smartwatch key press inference side channel attack.

\

References

[1] Maiti, A., Jadliwala, M., He, J. and Bilogrevic, I. (2018). *Side-Channel Inference Attacks on Mobile Keypads Using Smartwatches*. IEEE TRANSACTIONS ON MOBILE COMPUTING,. IEEE, pp.1-15.

[2]En.wikipedia.org.(2018). *Smartwatch*.[online] Available at: https://en.wikipedia.org/wiki/Smartwatch