I Know What You Type: Leaking User Privacy Via Novel Frequency-based Side-channel Attacks

Rui SONG*, Yubo SONG*, Shang GAO†, Bin Xiao† and Aiqun HU*
*School of Information Science and Engineering, Southeast University
†Department of Computing, The Hong Kong Polytechnic University
{songrui, songyubo}@seu.edu.cn, cssgao@comp.polyu.edu.hk, csbxiao@polyu.edu.hk, aqhu@seu.edu.cn

***Abstract*—Smartphone sensors have been applied to record the movement of users for healthy use. However, the motion sensor readings recorded by malicious applications can be utilized as a side-channel to leak user privacy by keystroke inference. Most existing approaches use time-domain statistical characteristics for keystroke inference. Their systems are poor to show the subtle changes in short time period, since the time-domain statistical features can only reflect the characteristics in a long-time interval.**

In this paper, we propose a novel framework to perform keystroke inference on smartphones. This framework introduces an improved MFCC algorithm to extract frequency-domain features for more comprehensive use of raw data. Since the frequency-domain energy distribution of motion signals is concentrated, and the specificity of signals is strong, MFCC can improve the inference accuracies under complex scenarios. Based on this framework, we present a prototype called FreqKey, which is an inference system to leak user privacy such as PINs and passwords. FreqKey collects motion sensor readings during keystroke events and constructs classification models with machine learning algorithms. Experimental results show that FreqKey improves the performance in a variety of complex scenarios. Especially, even in web platform whose sampling rate is lower than 80Hz, FreqKey can achieve relatively high accuracy of 74.6%. To mitigate the frequency-based side-channel attack and protect user privacy, we propose a defense solution which contains sensor-activity monitoring, malicious program identification and interference signal injection.

I. Introduction

Mobile sensors are embedded into smart mobile devices to provide users with more ways of interaction. Motion information like location, orientation and movement can be recorded by sensors like accelerometer, gyroscope and orientation sensor. However, vulnerabilities have been confirmed that data recorded by motion sensors can be utilized as a side-channel to steal user privacy by keystroke inference[1].

With the help of machine learning algorithms, researchers can infer user privacy from motion sensor readings[2]. Several native applications have been designed to infer the passwords[3], text messages[4] or unlocking gestures[5] of the devices from users' keystroke.

In addition to native applications, inference systems based on web applications can also be utilized for keystroke inference[6][7]. Compared to native applications, web applications can be executed on almost every mobile platform and be activated without downloading. With the help of cross-site scripting (XSS) techniques, malicious code can be injected to web pages and steal victims' privacy without their awareness.

However, attacks mentioned above also have great limitations. Since the using scenarios are more complex in real world, the inference accuracy of the attack systems will decrease significantly. This is because the inertial data recorded by motion sensors contains not only signals caused by users' keystroke, but also the disturbance caused by other movements of users.

Since most of the researches only extract time-domain statistic features from the raw data, they failed to obtain characteristics in frequency-domain, which can help distinguish between the data patterns of keystroke behavior and other movement in complex scenarios. Thus, systems built by previous researchers will get poor performance in real life scenarios.

Thus, we propose a novel framework based on features from both time-domain and frequency-domain to address the problem of accuracy decrease in complex scenarios. An algorithm called Mel Frequency Cepstral Coefficients (MFCC) is utilized to extract features from frequency-domain. This algorithm is based on cepstral analysis, which is a technique originally used in speech recognition and voice processing fields[8]. It has been proved that MFCC can distinguish the characteristics of complex signals and extract features from frequency-domain efficiently[9][10]. Appropriate improvements have been introduced to MFCC to satisfy our application scenarios.

Based on this framework, we present a novel prototype called FreqKey, which performs keystroke inference attack by analyzing motion sensor data with machine learning algorithms. By designing applications on different platforms, FreqKey can collect data of accelerometer, gyroscope and orientation sensor from different devices in different scenarios. By applying MFCC for feature extraction, FreqKey is able to extract features in frequency-domain, which can be utilized to distinguish between the motion signals of keystroke events and signals caused by other body and device movement.

The key contribution of our work is summarized as follows:

- We collect sensor data of keystroke events from volunteers and build an original data set. 300 participants are invited to enter numerical sequences into FreqKey. We collect more than 290,000 labelled motion sensor readings from Android and iOS devices, which come from 3,000 keystrokes in total.
- We propose a novel framework which introduces the cep-

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

stral analysis and MFCC algorithm for feature extraction to improve the inference accuracy of the system. Features extracted by MFCC algorithm can effectively distinguish between keystroke signals and the ambient noise.
- We design FreqKey, a novel prototype which is based on the framework above. FreqKey can steal user privacy such as PINs and passwords. In FreqKey, time-domain and frequency-domain features are extracted to construct feature sets which are then placed into classification algorithms to construct classifiers finally.

The rest of the paper is organized as follows. Section 2 presents the background and the related work in this field. Section 3 introduces the structure of the FreqKey system. Section 4 discusses the theory of cepstral analysis and MFCC algorithm. Section 5 introduces the evaluation on FreqKey from several dimensions. Section 6 reveals the countermeasures and mitigation strategies against keystroke attacks. Section 7 concludes the whole paper finally.

II. Related Work

Side-channel attack have been performed to break into the security protection of computer systems and other crypto-systems for a long time. Side-channels like power consumption[11], timing signals[12], electromagnetic leakage[13] and acoustic signals [14] can all be used to steal user privacy efficiently.

However, attacks on smartphones are slightly different. Most of the side-channel attacks on mobile phones tend to use mobile sensors for data collection. Early researches utilize sensors or hardware embedded in devices like camera[15], geolocation sensors[16] and microphones[17]. However, since the system architecture is fully developed these days, calls to these sensitive sensors are strictly limited.

Some of the experiences from smart wearable devices can be adapted to perform attacks on smartphones. Previous researches have confirmed that motion sensors can also be utilized as side-channels. Wang el al. use accelerometers embedded in smart watches to distinguish typing events on laptop keyboards and extract English words typed by victims[18]. Wang et al. use motion sensors to trace the moving trajectory of the users, and then infer the users input on ATM keypads or regular computer keyboards[19].

Given that the security mechanisms in smartphones have few limitations on motion sensors, they can be used as side-channels to steal user privacy. Researches have confirmed the feasibility of keystroke inference systems based on motion sensors. Cai et al. build an inference system based on data collected from orientation sensor[1]. They perform attacks on a number-only keypad and get 70% inference accuracy. Xu et al. use accelerometer for keystroke event detection and orientation sensor for position inference. They innovatively put the analysis progress on mobile phones, which greatly enhances the integration of their system[3].

In addition to the inference of numerical sequences, some researchers also design systems to steal text messages. Owusu et al. design a system called ACCessory, which can infer the character input and keystroke position under different screen region granularities[2]. Ping et al. focus on long text input like tweets or emails. They introduce the techniques of natural language processing to improve the inference accuracy to 65%[20].

There are also systems based on web applications which can be executed on multiple platforms. Mehrnezhad et al. design a system called PINlogger on web platform. This system can steal victims' motion sensor data when they active the malicious pages actively or passively[21].

Given the existing studies, we find that most researches only extract time-domain statistical features from original signals. Their work can get quite good results under ideal conditions but poor performance in complex scenarios[22]. Meanwhile, frequency-domain features can reveal the dynamic characteristics of signals, which can help to distinguish the keystroke signals from ambient noise. Thus, we introduce MFCC for frequency-domain feature extraction, which can increase the inference accuracy in complex scenarios.

III. Implementation

In this section, the implementation of FreqKey is revealed by steps, and the key problem to be addressed by the system is also shown in detail. The FreqKey system can be divided into 3 parts: data collection, feature extraction and model training.

A. Data Collection

To evaluate the performance of FreqKey in multiple scenarios, we design native applications for Android and iOS systems and a web application for cross-platform experiment. These applications collect sensor readings from accelerometer, gyroscope and orientation sensor when keystrokes are performed. The sensor readings are then transmitted to the server with the corresponding numbers as the classification labels in real time.

The server receives the labelled readings and store them into the database. Given that this system will face high-frequency concurrent requirements and complex I/O requests, we construct the server with Node.js, which is a lightweight JavaScript runtime environment based on an event loop mechanism. This mechanism utilizes the asynchronous features to prevent I/O blocking and can address high-concurrency requirements.

B. Feature Extraction

Features of the original data should be extracted before model training. As mentioned before, features from time-domain and frequency-domain are extracted respectively.

The raw data which comes from 3 sensors can be divided into 4 signal sequences: acceleration with gravity, acceleration without gravity, rotation speed and orientation parameter. Each of the sequence above have 4 axes including an extra virtual axis: the Euclidean norm of each sequence. The Euclidean distances are extracted to reveal the energy or power received by the devices.

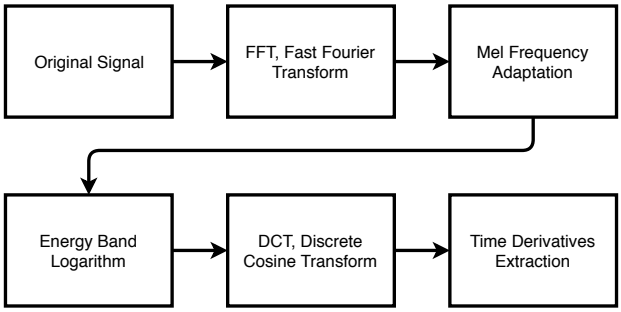

Fig. 1: Diagram of MFCC extraction.

TABLE I: Features extracted from time-domain.

Feature	Description
Max	Maximum value
Min	Minimum value
RMS	Root-mean-square value
RMSE	Root-mean-square error
Mean	Average value of each tap
PeakNum	Number of local peaks
TroughNum	Number of local troughs
SMA	Signal magnitude area
Skewness	Asymmetry of the curve
Kurtosis	Peakedness of the curve
ATP	Average time to a peak
ATT	Average time to a trough

In time-domain, several statistical features listed in TABLE I are extracted. 192 time-domain features are extracted from each keystroke event finally. In frequency-domain, the raw data is first transformed into frequency spectrum by Fast Fourier Transform. Then the signals are adapted into cepstral analysis and MFCC algorithm whose progress will be presented in Section 4 in detail.

C. Model Training

Several classification algorithms are utilized in training part to construct classifiers based on the features extracted from time-domain and frequency-domain. The algorithms we used is listed below.

- Bayesian Network
- Support Vector Machine(SVM)
- Bagging (Bootstrap Aggregating)
- Artificial Neural Network(ANN)
- Logistic Model Tree(LMT)
- Logit Boost
- Random Forest

The classification algorithms come from different fields, whose basic theories are also different from each other. This ensures that we can comprehensively evaluate the robustness of the entire system from multiple perspectives and select the classification algorithm which is most suitable for this task.

IV. Mel Frequency Cepstral Coefficients

A. Justification

In FreqKey, MFCC, which is a technique widely used in voice signal processing and blind signal separation area, is introduced to extract features in frequency-domain. Since there are several similarities between voice signals and motion signals, this algorithm can be adapted for feature extraction from motion signals[23]. Although the frequency range of voice signals and motion signals are different, these signals share similar characteristics in their respective frequency-domains. And the energy of the signals are both concentrated in low frequencies. In addition, both voice signals and motion signals are hard to reproduce, which means the features extracted from these signals must be robust to resist the uncertainty of the original data.

After considering several popular techniques in voice signal processing area, MFCC is selected for feature extraction. MFCC algorithm is originally used to distinguish between the formants and the detail of signals, while formants and envelope of the signals can reflect the primary information and the detail reflects the ambient noise. By arranging a set of bandpass filters according to the size of the critical bandwidth from low frequency to high frequency, MFCC can filter the input motion signals and keep the signal components caused by keystroke as much as possible.

B. Cepstral Analysis

Peaks, which are usually called as formants, can be used to distinguish the characteristics of signals. To get the formants, spectral envelope is extracted because it contains the positions of formants and the changing process of the formants.

IFFT is performed on original spectrum to separate the spectral envelope from the spectrum. IFFT of the original spectrum can be considered as a new pseudo-frequency domain, in which the frequency of envelope is obviously lower than the frequency of the noise. By passing the signal through a low-pass filter, the spectral envelope can be extracted from the original spectrum.

C. Implementation of MFCC

Fig. 1 shows the whole process of MFCC extraction.

Fast Fourier Transform is performed to get the frequency spectrum of each frame of the raw data. And then the linear

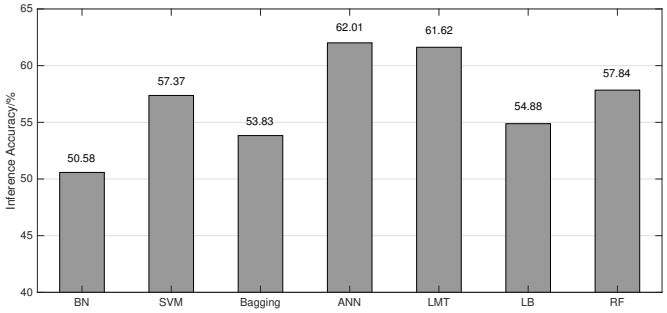

Fig. 2: Accuracies with features of time-domain only.

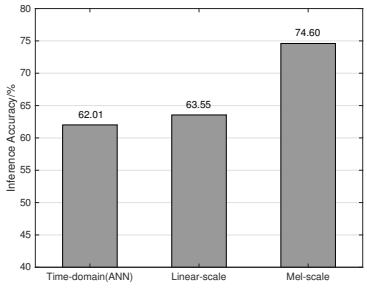

Fig. 3: Accuracies with frequency-domain features extracted in different scales.

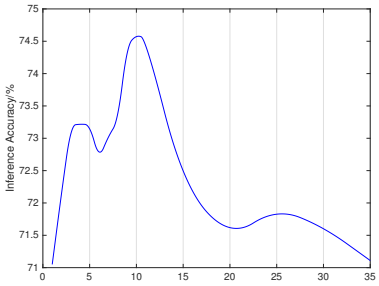

Fig. 4: Accuracies when extracting different numbers of MFCCs.

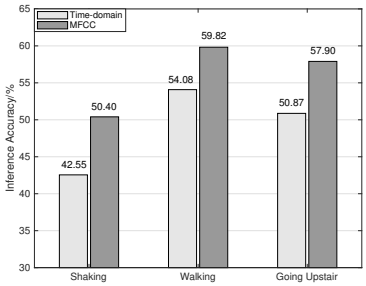

Fig. 5: Accuracies when users are in motion states.

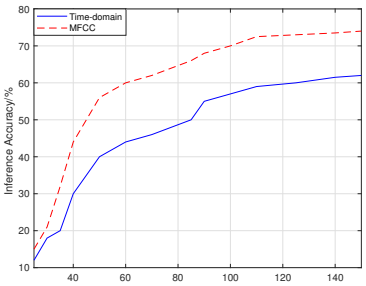

Fig. 6: Accuracies under different sampling rates.

scale is converted to Mel scale, which is a non-linear frequency scale based on the frequency distribution of motion signals. The signals after scale conversion will be passed through a set of filters, whose distribution is dense in low frequencies and sparse in high frequencies. After that, the energy band logarithm is computed to get the logarithm of the former spectrum for cepstral analysis.

Next step is to calculate the Discrete Cosine Transform. DCT is used in MFCC since it can get frequency-domain features which do not contain imaginary part and can further compress the results to get efficient features from fewer coefficients. The standard MFCCs can only represent the static characteristics of signals. Since the motion sensor data is continuous in time-domain, dynamic differential parameters can be calculated to reflect the time-domain continuity characteristics of the original data.

V. Evaluation

In this section, the performance of FreqKey is evaluated by a series of experiments from different aspects. 300 participants were invited to participant in the experiments following our instructions. All the volunteers are college students and perennial smartphone users. The participants were instructed to open the application and enter the number displayed on the prompt box. The numbers provided by the prompt box were random sequences to avoid the similar patterns of sensor data from fixed sequences. The experiments collected 300 groups of data of numerical sequences, which correspond to 3,000 keystroke events. About 290,000 raw labelled sensor readings were collected from accelerometer, gyroscope and orientation sensor during the whole process.

A. Performance in Only Time-domain Features

Machine learning techniques introduced in section 3 are used to construct classification models. In this phase, classifiers are trained by the feature sets extracted from time-domain. The features to be extracted have been introduced in TABLE I. 10-fold cross-validation is introduced to take full advantage of limited raw data. Fig. 2 shows the performance of each algorithm in the case of time-domain feature input only. The result shows that ANN get the best inference accuracy in both platforms, and the performance of LMT algorithms is relatively good, only slightly inferior to ANN. However, the results of Bayesian Network and Bagging algorithms indicate that these algorithms are obviously not suitable for this classification work.

B. Performance after Adding Frequency-domain Features

MFCCs are extracted as the frequency-domain features, which can help to distinguish between signals caused by keystroke events and other movement. Fig. 3 shows the inference accuracy under different scales. It indicates that adding simple linear-scale frequency-domain features can slightly improve the performance. And MFCC features which extracted from Mel-scale can improve the performance more efficiently, which can be up to 74.60%.

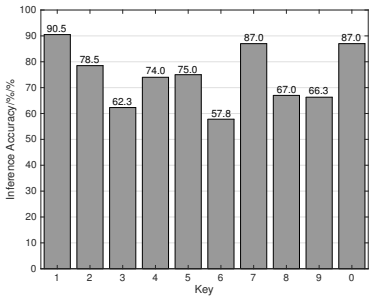

Fig. 7: Accuracy of keys on iOS devices.

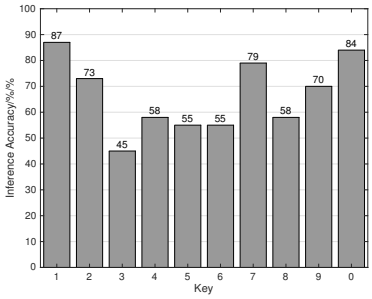

Fig. 8: Accuracy of keys on Android devices.

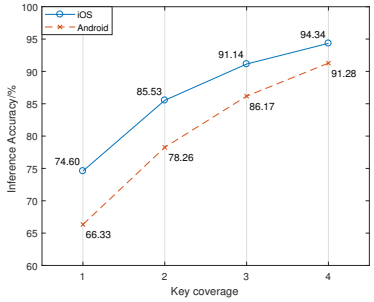

Fig. 9: Accuracy when cover different digits.

The number of coefficients extracted from each frame should also be considered, since this is related to the size of feature sets, which would greatly affect the efficiency of machine learning algorithms. Fig. 4 shows the inference accuracy when extract different coefficients from one frame. The result shows that the best performance comes when there are 10 coefficients extracted from each frame.

C. Performance in complex scenarios

Several applications are designed to evaluate the performance of FreqKey in complex scenarios. Fig. 5 shows the performance when the users perform keystroke events in motion states. The candidates are instructed to click the buttons while shaking their bodies, walking or going upstairs to reflect the actual situations in the daily lives. The results show that if only taking time-domain features into account, the performance of FreqKey will decrease drastically when people perform keystroke events in motion. However, after introducing MFCC for frequency-domain feature extraction, the performance will be improved obviously.

The relationship between sampling rate of motion sensors and the inference accuracies has been discussed by researches[2]. Since MFCC algorithm can extract more efficient information from the original signals, it is meaningful to investigate the performance in low sampling rate scenarios. Fig. 6 indicates the trend of the inference accuracies under different sampling rate before and after introducing MFCC. The result shows that MFCC extraction can improve the performance under the low sampling rate scenarios.

D. Impact of Key Position to Inference Accuracy

It is necessary to emphasis the distribution of inference accuracy of different keys. Fig. 7 shows the accuracies of each key on iOS devices. The result indicates that keys on the edges of the keypad always get higher accuracy rates than those of inner area. The result also indicates that keys on the right side of the keypad always get lower accuracy rates than those on the left side. Since most of the volunteers invited by us are right-handed, they usually use their right hands to hold the devices and perform keystrokes with the right thumbs. To make it easier to reach the buttons on the left side, they should swing the devices more violently. The result on Android devices is similar, as shown in Fig. 8.

TABLE II: Ranked key table, three most confused numbers are listed for each key.

Key	True Value	2nd	3rd	4th
1	90.5	**4**: 7.5	**2**: 2.0	-
2	78.5	**5**: 8.5	**1**: 3.0	**3**: 2.0
3	62.3	**6**: 14.1	**9**: 6.5	**8**: 5.5
4	74.0	**1**: 12.0	**2**: 7.0	**7**: 3.0
5	75.0	**2**: 10.0	**8**: 5.0	**3**: 3.0
6	57.8	**9**: 22.1	**3**: 13.1	**2**: 2.5
7	87.0	**4**: 4.0	**0**: 2.5	**1**: 2.0
8	67.0	**6**: 8.5	**9**: 6.0	**3**: 5.5
9	66.3	**6**: 16.6	**8**: 8.5	**3**: 6.5
0	87.0	**8**: 6.5	**9**: 2.5	**7**: 2.0

The confusion matrix shows that the wrong inference key positions tend to gather around the correct keys. Most of the inferences come together on the 3 to 5 adjacent keys around the correct key. This means that the false inference samples can also be utilized to provide redundant information. Thus, a ranked key table can be built to reduce the searching space. TABLE II shows a ranked key table based on the result of ANN learning with MFCC features. The table lists the most probable inferred positions for each key. Fig. 9 indicates the potential ability when take more digits into consideration. When improving the potential target digits to 4, the total accuracy of the system can be improved to about 94.34% on iOS devices and 91.28% on Android devices.

VI. Mitigation Strategy

Given the danger of keystroke attacks, it is necessary to consider targeted countermeasures. However, separating malicious sensor calls from benign application is not easy. Traditional strategies typically reduce the sampling rate of all background programs, which would affect the use of benign applications. However, we find that the pattern of sensor signals from malicious programs has certain characteristics. Thus, pattern recognition can be utilized to identify malicious applications. According to this, we propose a countermeasure solution against this kind of attack.

A. Sensor Activity Monitoring

We find that sensor calls from benign applications are usually short-lived while malicious programs tend to monitor a fixed combination of sensors over a long period. And the progress of recording sensor data is usually accompanied by data transmission through Internet in malicious applications. Thus, the state of sensors can be recorded by a monitoring program, which can record sensor activities based on sensor combination, active duration and network usage.

B. Malicious Program Identification

By extracting features from the data collected by the monitoring program, pattern recognition can be utilized to distinguishing the malicious program from massive sensor calls. Pattern recognition techniques can detect appropriate features which can contribute to identify the malicious calls. Then these feature sets can be aggregated for machine learning. Classification algorithms can be utilized to distinguish malicious code from massive sensor calls.

C. Interference Signal Injection

Defense system based on noise injection has been proposed by researchers to mitigate keystroke attacks[24]. However, the existing countermeasure is mainly focus on systems which only extract time-domain features. To make this method valid in systems based on frequency-domain features, noise whose pattern is similar to signals caused by keystroke events can be injected and confuse the inference systems.

VII. Conclusion

In this paper, we construct a system called FreqKey to inference numerical input on smartphones by the data from motion sensors. To collect data from different platforms, we build applications which can record the motion sensor data when users perform keystroke events. To adapt our attack system to more complex scenarios, we introduce MFCC algorithm to distinguish between signals caused by keystroke events and other device movements. The evaluation of FreqKey shows that MFCC algorithm can improve the one-phase accuracy rate to 74.60% comparing to the former rate of 62.01%. Based on the result of our research, we propose a solution against this attack system, which can mitigate this attack and protect the privacy of users.

References

[1] L. Cai and H. Chen, "Touchlogger: inferring keystrokes on touch screen from smartphone motion," in *Usenix Conference on Hot Topics in Security*, 2011, pp. 9–9.

[2] 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.

[3] Z. Xu, K. Bai, and S. Zhu, "TapLogger: inferring user inputs on smartphone touchscreens using on-board motion sensors," in *ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2012, pp. 113–124.

[4] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tapprints: your finger taps have fingerprints," in *International Conference on Mobile Systems, Applications, and Services*, 2012, pp. 323–336.

[5] M. Hussain, A. Al-Haiqi, A. Zaidan, B. Zaidan, M. Mat Kiah, N. B. Anuar, and M. Abdulnabi, "The rise of keyloggers on smartphones: A survey and insight into motion-based tap inference attacks," *Pervasive and Mobile Computing*, vol. 25, pp. 1–25, Jan. 2016.

[6] R. Song, Y. Song, Q. Dong, A. Hu, and S. Gao, "Weblogger: Stealing your personal pins via mobile web application," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct 2017, pp. 1–6.

[7] M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao, "Touchsignatures: identification of user touch actions and pins based on mobile sensor data via javascript," *Journal of Information Security and Applications*, vol. 26, pp. 23–38, 2016.

[8] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *arXiv preprint arXiv:1003.4083*, 2010.

[9] M. A. Rao and P. K. Ghosh, "Pitch prediction from mel-frequency cepstral coefficients using sparse spectrum recovery," in *Communications (NCC), 2017 Twenty-third National Conference on*. IEEE, 2017, pp. 1–6.

[10] P. Prajapati and M. Patel, "Feature extraction of isolated gujarati digits with mel frequency cepstral coefficients (mfccs)," *International Journal of Computer Applications*, vol. 163, no. 6, pp. 29–33, 2017.

[11] H. Lin, A. Slagell, Z. Kalbarczyk, P. Sauer, and R. Iyer, "Runtime semantic security analysis to detect and mitigate control-related attacks in power grids," *IEEE Transactions on Smart Grid*, 2017.

[12] Z. H. Jiang, Y. Fei, and D. Kaeli, "A complete key recovery timing attack on a gpu," in *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 394–405.

[13] J. Hao, Y. Gong, L. Jiang, and J. Fan, "Analytical formulation for electromagnetic leakage field to transmission line coupling through covered apertures of multiple enclosures," *Advances in Materials Science and Engineering*, vol. 2017, 2017.

[14] A. Faruque, M. Abdullah, S. R. Chhetri, A. Canedo, and J. Wan, "Acoustic side-channel attacks on additive manufacturing systems," in *Proceedings of the 7th International Conference on Cyber-Physical Systems*. IEEE Press, 2016, p. 19.

[15] N. Xu, F. Zhang, Y. Luo, W. Jia, D. Xuan, and J. Teng, "Stealthy video capturer: a new video-based spyware in 3g smartphones," in *Proceedings of the second ACM conference on Wireless network security*. ACM, 2009, pp. 69–78.

[16] L. Cai, S. Machiraju, and H. Chen, "Defending against sensor-sniffing attacks on mobile phones," in *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*. ACM, 2009, pp. 31–36.

[17] R. Schlegel, K. Zhang, X. Zhou, M. Intwala, and A. Kapadia, "Soundminer: A stealthy and context-aware sound trojanfor smartph-ones."

[18] H. Wang, T. T.-T. Lai, and R. Roy Choudhury, "Mole: Motion leaks through smartwatch sensors," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 155–166.

[19] C. Wang, X. Guo, Y. Wang, Y. Chen, and B. Liu, "Friend or Foe?: Your Wearable Devices Reveal Your Personal PIN." ACM Press, 2016, pp. 189–200.

[20] D. Ping, X. Sun, and B. Mao, "TextLogger: inferring longer inputs on touch screen using motion sensors." ACM Press, 2015, pp. 1–12.

[21] M. Mehrnezhad, E. Toreini, S. F. Shahandashti, and F. Hao, "Stealing pins via mobile sensors: actual risk versus user perception," *International Journal of Information Security*, pp. 1–23, 2016.

[22] 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.

[23] L. Liu, M. Popescu, M. Skubic, M. Rantz, T. Yardibi, and P. Cuddihy, "Automatic fall detection based on doppler radar motion signature," in *2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 2011, pp. 222–225.

[24] P. Shrestha, M. Mohamed, and N. Saxena, "Slogger: Smashing motion-based touchstroke logging with transparent system noise," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 67–77.