# VibLeak: Towards Practical Covert Data Leakage via Phone Vibrations

Junye Jiang[12], Jingcheng Ju[12], Haowen Xu[3], Zhenlu Tan[12], Duohe Ma[12*], Jun Dai[3], Xiaoyan Sun[3*]
[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Department of Computer Science, Worcester Polytechnic Institute, Massachusetts, USA
{jiangjunye, jujingcheng}@iie.ac.cn, {hxu4}@wpi.edu, {tanzhenlu, maduohe}@iie.ac.cn, {jdai, xsun7}@wpi.edu

***Abstract*—Privacy leakage is a growing concern in smartphone security. Previous studies demonstrated the feasibility and limitations of data transmission via vibration using customized devices under ideal conditions, but focused mainly on transmission speed. Through the analysis of real-world smartphone usage scenarios, we have found that there is a potential risk of private user data on Android phones being actively and covertly leaked because of the poor management of their built-in motion sensors. This paper introduces VibLeak, a novel covert-channel attack framework that intentionally leaks data through vibration. We developed a malicious app to implement this framework and conducted comprehensive experiments across various Android smartphones and environments. The results reveal that VibLeak can transmit data with remarkable accuracy and speed even under realistic conditions, employing vibration intensity that is imperceptible to most users. Our work not only uncovers this previously overlooked privacy leakage vector but also underscores the critical need for advanced security measures to address such sophisticated threats in the evolving landscape of smartphone technology.**

***Index Terms*—Mobile Security, Data Leakage, Covert-channel Attack**

## I. Introduction

With smartphones becoming the primary carrier of sensitive data, privacy and security issues are becoming increasingly prominent. Since users have to authorize local file access permissions in order to use application functions normally, this give attackers opportunities to steal private data. However, even if attackers can read files, illegal external transmission of sensitive files is restricted or prohibited due to the strict censorship mechanism and system privacy protection framework [1] of the application store.

The built-in motion sensors in smartphones, such as accelerometers and gyroscopes, can accurately measure subtle movements of the phone and are widely used in fitness tracking applications and motion sensing games. Some works [2]–[10] attempt to steal user privacy information by exploiting the loose regulation of motion sensor data. However, these works of privacy information steal passively wait for vibrations to occur. We found that sensitive files can be actively converted into rhythmic vibrations through built-in motors of phones, and then recorded as insensitive vibration data through motion sensors and transmitted legally later.

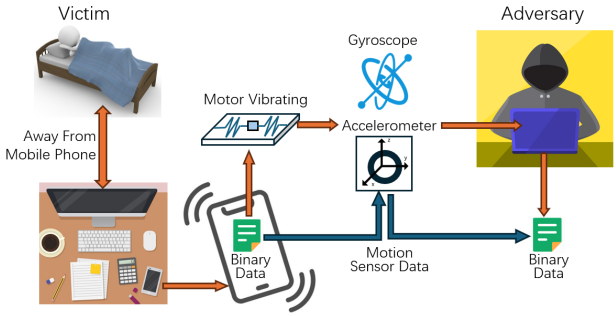



Fig. 1: Leakage of mobile phone data through vibration without the victim's awareness.

Some works had already shown that data can be transmitted by vibrations. Hwang et al. [11] first attempted to use motor and accelerometer on tables to transmit small amounts of data. Yonezawa et al. [12] shortened the interval between each vibration by performing special treatment on the residual vibration after each vibration. Roy et al. [13], [14] designed a special vibration generating device that uses signal modulation and encoding to map binary data to vibrations of different frequencies, greatly improving data transmission speed. At the same time, they also analyzed the inability to achieve vibration frequency modulation due to hardware limitations of mobile phone motors. Lee et al. [15] used a high sampling rate accelerometers to quickly transmit data by combining multiple consecutive binary bits. Sen et al. [16] designed a ring that can generate vibration, providing a key exchange scheme for secure communication between IoT devices and users' smartphones through vibration. Wijewickrama et al. [17] innovatively used homemade equipment to transmit vibrations through human skin, which has strong concealment but average effectiveness. Cui et al. [18] abandoned traditional accelerometers and gyroscopes and introduced millimeter wave radar as a vibration receiving device, which can also be used in conjunction with smartphone motors to achieve medium distance data

*Corresponding authors: Duohe Ma and Xiaoyan Sun.

transmission.

The main purpose of the above works is to evaluate the feasibility of using vibration for data transmission in various devices and scenarios. In contrast, our research focuses on using unmodified built-in hardware for data transmission in a single smartphone, and we pay more attention to the concealment of data transmission. Therefore, we propose a new covert-channel attack framework VibLeak. We make smartphones actively vibrate to leak any available user data, rather than passively collecting vibration data generated by voice. Since in Android systems both motion sensors and the motors used to generate vibrations do not require specific permissions for use, we only focus on Android phones. Figure 1 shows the structure of VibLeak.

The major contributions of this research are summarized as follows:

- We proposed a covert-channel attack framework for leaking private data from smartphones, which utilizes the built-in motor of the phone to convert any sensitive files into a vibration signal, generating accelerometer readings that can be transmitted externally without being noticed or prohibited. Once the accelerometer readings are received, the original sensitive files can be restored by denoising, segmenting, and recognizing the accelerometer readings.
- We evaluated the effectiveness of VibLeak in various scenarios by assessing different impact factors such as phone models, amplitude, vibration duration, phone status, etc.
- Due to VibLeak's goal of leaking private data secretly, we did experiments focusing on the concealment of vibrations, thereby offering practical solutions for conducting these attacks.

## II. VibLeak

### A. Threat Model

We assume that the victim user possesses a smartphone equipped with motion sensors and a motor. We crafted a malicious application disguised as a sports or health application that can record users' running or other data, provide cloud storage capabilities, and conduct behavioral analysis by uploading sensor data. It incorporates three benign functionalities: reading files, control motor and motion sensors, and transmission. With the help of this app, as illustrated in Figure 2, VibLeak consists of two stages. In the first stage, we read the binary format of the file and determine the vibration operation based on the value of each bit. At the same time, we use motion sensors to record vibrations and upload the recorded data after transmission is complete. In the second stage, after obtaining the recorded data, we perform denoising, segmentation, and recognition operations on the data, and then restore the original file.

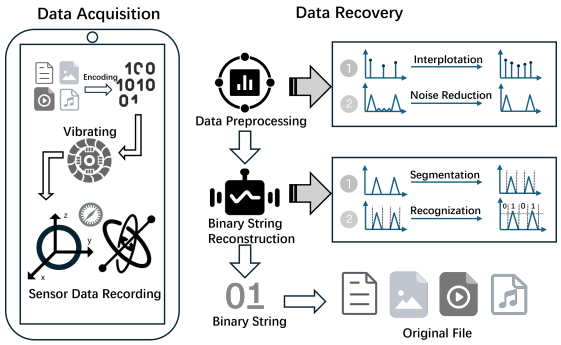



Fig. 2: System architecture of VibLeak.

### B. Convert Binary Data to Vibration Sequences

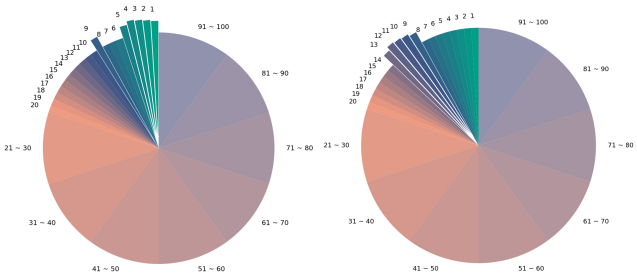
The conversion of file data to mobile phone vibration sequences requires that the binary '1' and '0' are able to match the corresponding states of vibration and non-vibration of the mobile phone. We define $T$ (*bit duration*) as the time duration allocated for each bit, i.e., to transmit $\frac{1}{T}$ bits of data per second. After a series of tests, we determined that $T = 100$ ms can balance transmission speed and accuracy without additional tampering with the phone. Specifically, when the binary is '0', the motor is stationary for $T$=100 milliseconds, whereas when the binary is '1', the 100 milliseconds are split into three distinct time periods: $t_1$, $t_2$ and $t_3$, respectively representing stationary, vibration (*vibration duration*), and stationary.

### C. Data Preprocessing

Given the low sampling rate and irregular sampling durations of the accelerometer, we initially apply frame interpolation to the collected data. This process enhances the data resolution, resulting in a smoother signal, and converts the unevenly sampled data into a uniformly sampled format.

Furthermore, the accelerometer readings are subject to interference from white noise and vibrational noise, necessitating the filtering of the interpolated data. We start this process by conducting a Fast Fourier Transform (FFT) analysis of the sensor readings under both stationary conditions and during periods of vibration. As illustrated in Figure 3, we have segmented the frequency range from 0 to 500 Hz into 100 durations of 5 Hz each, and we have identified and labeled the five frequency domain durations exhibiting the highest amplitudes. Among them, the vibration point frequencies are concentrated in groups of 9 to 13, i.e., 45 to 65 Hz, while the non-vibration point frequencies are concentrated in groups of 1 to 5, i.e., 5 to 25 Hz.

Therefore, we initially implement a high-pass filter on the data to eliminate frequencies below 35 Hz, thereby effectively mitigating the majority of low-frequency noise interference. Subsequently, we employ the Wiener filter to further attenuate background noise and improve the distinguishability between vibration and non-vibration data segments.



(a) No Vibration (b) Vibration

Fig. 3: FFT results across different states.

### D. Binary String Reconstruction

VibLeak is inspired by the premise that human perception of vibration at a given point is influenced by the rate of amplitude change relative to adjacent points. This feature can be represented by standard deviation($\sigma$). For each point considered, we define a small range centered around it and calculate the standard deviation as the amplitude change rate of that point. Subsequently, we calculate the standard deviation of all data ($array$) and multiply it by an empirical coefficient as a threshold ($threshold1$). We can use this threshold to determine whether a point is in a vibrating state and find the first vibrating point ($index_{start}$).

Next, we obtain the range of points of the first vibration and the index of the $N$ points with the largest amplitude by using $index_{start}$and the time required for each vibration operation ($pertime$). Finally, we calculate the average index of the maximum and minimum points among $N$ points as the midpoint of the first vibration. At the same time, we update the starting index of the first vibration ($index_{start}$). The overall process is shown in Algorithm 1.

Then, based on the time point at which the current vibration operation ends and the time required for each vibration operation, the time range for all subsequent vibrations can be derived. We calculate the average amplitude of each point within the vibration range and compare it with another threshold ($threshold2$) to determine whether to output "0" or "1". This threshold is dynamically obtained through a comprehensive experimental process.

## III. Experiments

### A. Mitigating Unstable Bit Duration

The design of the Android system results in kernel scheduling algorithms not prioritizing strict timing control, so when developers attempt to allocate specific durations in applications, actual execution may exhibit instability.Therefore, it is impractical to impose strict control over the duration of individual binary bits at the software level within the Android environment. As illustrated in Figure 4, we conducted tests to measure the duration for a bit under both vibrational and non-vibrational conditions. When the designated bit duration is set at 100 milliseconds, the actual

**Algorithm 1** Find start point based on standard deviation

**procedure** FindStartPoint($array$, $range$, $pertime$, $N$)
  $threshold1 \leftarrow \sigma(array) \div 2$
  **for** $i \leftarrow 0$ to length$(array) - 1$ **do**
    **if** $\sigma(array[i-range, i+range]) > threshold1$ **then**
      $index_{start} \leftarrow i - pertime \div 2$
    **end if**
  **end for**
  $array_{sub} \leftarrow abs(array[index_{start}, index_{start} + pertime])$
  $indexes_{topN} \leftarrow$ GetTopNIndexes$(array_{sub}, N)$
  $index_{max} \leftarrow max(indexes_{topN})$
  $index_{min} \leftarrow min(indexes_{topN})$
  $index_{average} = (index_{max} + index_{min}) \div 2$
  $index_{start} = index_{average} - pertime \div 2 + index_{start}$
**end procedure**

duration consistently exceeds this threshold in both scenarios, exhibiting significant variability. In order to improve the stability of bit duration, we use a strategy to set the end time of each bit duration. This approach not only ensures that the execution time of each thread approximates the intended duration but also allows for dynamic adjustments to the time allocated for inter-thread scheduling, thereby ensuring adherence to the overall planned duration.

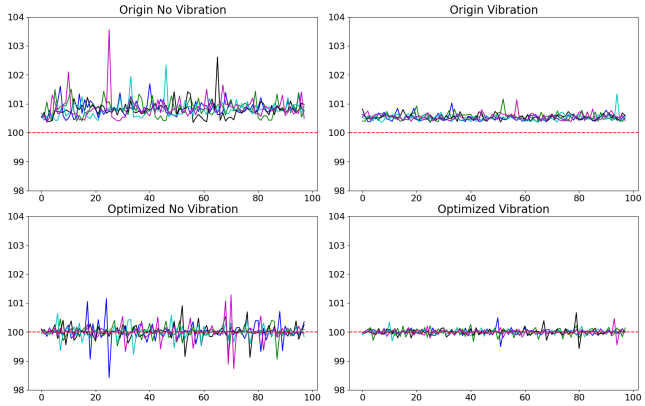



Fig. 4: Time per round: before vs. after optimization. Colored lines represent multiple test runs. Red dashed line: 100ms control threshold.

### B. Universality of Different Phones

In order to explore the universality of VibLeak, we conducted experiments on mobile phones with different motor types and hardware positions. We selected ten Android phones from different brands and models as experimental equipment, set the bit duration of each vibration operation to 100 milliseconds, the vibration duration to 15 milliseconds,

TABLE I: Universality of different phones

| Device | Sensor Axis | Sampling Rate | Motor | LRA | BRA |
|---|---|---|---|---|---|
| Xiaomi 14 | Z | 200Hz | X-axis Linear Motor | 100% | 99.31% |
| Xiaomi 14Pro | Z | 200Hz | X-axis Linear Motor | 100% | 99.08% |
| Xiaomi 15 | Y | 200Hz | X-axis Linear Motor | 100% | 98.89% |
| Redmi K50Ultra | X | 200Hz | X-axis Linear Motor | 100% | 98.94% |
| Redmi K70 | Y | 200Hz | X-axis Linear Motor | 90% | 95.10% |
| Huawei Mate 30pro | Z | 500Hz | X-axis Linear Motor | 100% | 99.84% |
| Honor 80GT | Z | 200Hz | Z-axis Linear Motor | 100% | 97.30% |
| Honor 60Pro | / | 200Hz | Rotating Motor | / | 63.27% |
| Realme Q3s | / | 200Hz | Pancake Motor | / | 53.25% |
| Oppo A92s | / | 400Hz | Rotating Motor | / | 54.35% |

and the amplitude to 255. For the experimental data, we used English text containing 100 bytes and consistently collected data using accelerometers. Each device has undergone 20 repeated tests to ensure the reliability and statistical significance of the results. To evaluate experiment results, we declare two primary metrics: Length Recognition Accuracy (LRA) and Binary Recognition Accuracy (BRA). LRA assesses the accuracy between detected length and actual length of the binary string , whereas BRA assesses the correctness of each bit in the recognized binary string. The results are presented in Table I.

It is easy to find that the transmission and restoration of vibrations in mobile phones with linear motors yield exceptionally high accuracy in both LRA and BRA. Notably, the LRA and BRA of Redmi K70 are lower compared to other phones. We analyzed the data and found that one set of data identified an extra bit at the end. Generally, length recognition error is mainly caused by inaccurate identification of the initial or final vibration. Actually, neither of these situations will affect the attacker's understanding of the recovery results. After excluding the abnormal data, the BRA of Redmi K70 exceeds 99%. The vibration intensity exhibited by nonlinear motors is much weaker than that of linear motors, making it difficult for motion sensors to sample the vibration. As a result, we cannot determine the duration of data transmission, and the BRA is also very low. Figure 5 shows the actual effect of transmitting images.

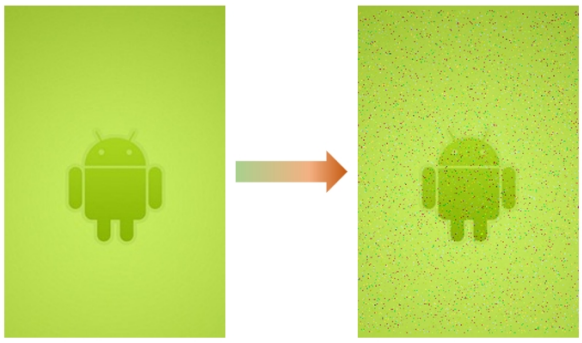


Fig. 5: Image transmission: original (left) vs. restored (right). Binary recognition accuracy: 98%.

### C. Amplitude Experiment

The amplitude of motor vibrations has a direct impact on the readings obtained from accelerometers. An increase in amplitude correlates with higher sensor readings, thereby facilitating the differentiation between vibrational and non-vibrational states. However, elevated vibration amplitudes also enhance the likelihood of being detected by observers. Consequently, achieving a successful attack necessitates a careful balance between recognition accuracy and stealth.

TABLE II: Amplitude experiment results

| Amplitude | ACC LRA | ACC BRA | GYR LRA | GYR BRA |
|---|---|---|---|---|
| 1 | 65% | 97.21% | 85% | 94.02% |
| 31 | 50% | 96.12% | 60% | 94.40% |
| 63 | 70% | 98.71% | 70% | 96.92% |
| 95 | 85% | 98.43% | 80% | 98.38% |
| 127 | 70% | 98.89% | 85% | 98.17% |
| 159 | 90% | 99.04% | 95% | 98.01% |
| 191 | 90% | 99.47% | 95% | 98.05% |
| 223 | 95% | 99.64% | 95% | 98.28% |
| 255 | 100% | 99.70% | 90% | 98.75% |

The Android vibration function allows adjusting the vibration parameters with amplitude values from 0 to 255 representing the intensity of the vibration from minimum to maximum. In our experiment, we utilized Xiaomi 14 smartphones to conduct experiments involving nine distinct amplitude settings, each with a vibration duration of 15 milliseconds. Data collection was performed using accelerometers(ACC) and gyroscopes(GYR), resulting in 20 data sets for each amplitude configuration, with each set comprising 50 binary sequences ("01"). We evaluated the algorithm's average recognition success rate and local recognition accuracy across varying amplitudes. The findings of the experiments are presented in Table II. The reason why the accelerometer performs better than the gyroscope in the experiment is because our algorithm is optimized for accelerometer data.

Analysis of the experiment results indicates that stronger vibrations yield higher accelerometer readings on mobile devices, which increases the algorithm's recognition accuracy. Notably, it was observed that when the amplitude exceeds 95, the improvements in Length Recognition Accuracy (LRA) and Binary Recognition Accuracy (BRA) become marginal.

### D. Vibration Duration Experiment

In addition to the amplitude of vibration, which influences the intensity of a single vibration, the vibration duration is also a critical factor affecting its intensity. When the amplitude is held constant, an increase in vibration duration correlates with an increase in vibration intensity. We established seven different vibration durations with amplitudes fixed at 255. The experimental data and equipment are consistent with those in Section III-C. The results of the experiment are shown in Table III.

TABLE III: Vibration duration experiment results

| Vibration Duration | ACC LRA | ACC BRA | GYR LRA | GYR BRA |
|---|---|---|---|---|
| 1ms | 75% | 96.32% | 40% | 83.41% |
| 5ms | 90% | 99.57% | 85% | 98.28% |
| 10ms | 100% | 98.82% | 85% | 98.55% |
| 15ms | 100% | 99.90% | 90% | 99.89% |
| 20ms | 100% | 99.78% | 95% | 99.90% |
| 25ms | 100% | 100% | 100% | 99.18% |
| 30ms | 100% | 100% | 95% | 99.53% |

An analysis of the experimental results revealed that maintaining an amplitude of 255 and a vibration duration of at least 10 milliseconds yields a length recognition accuracy of 100%. Furthermore, binary recognition accuracy can also reach 100% when the vibration duration is extended to 25 milliseconds. In contrast, when the vibration duration is reduced to 5 milliseconds, there is a slight decrease in length recognition accuracy; however, remarkably high binary recognition accuracy is still attainable. Additionally, the trend observed in the gyroscope data corresponds closely with that of the accelerometer data.

### E. Phone Status Assessment and Effectiveness

From the perspective of concealment, the victim needs to not touch their phones. To achieve this, it is essential to accurately discern the behavioral patterns of users, enabling the implementation of data transmission at specific time while the user are not in contact with the device for an extended period. We can directly determine the user's states through accelerometer readings. We designed an experiment to record the accelerometer readings of users in several common states using an accelerometer. Through comparative analysis of these data, we had the following observation: the sensor readings are markedly elevated when the phone is in motion or when the user is actively manipulating the device, compare to when the phone remains stationary and without physical contact with users. This finding offers a straightforward and effective determination mechanism: by analyzing fluctuations in sensor readings, we can accurately tell whether the phone is in a proper state for generating vibrations and transmitting data without drawing the user's attention. The comparative analysis of sensor readings across various scenarios is illustrated in Figure 6.

Following the determination that the stationary state of a mobile device is optimal for data transmission, we further

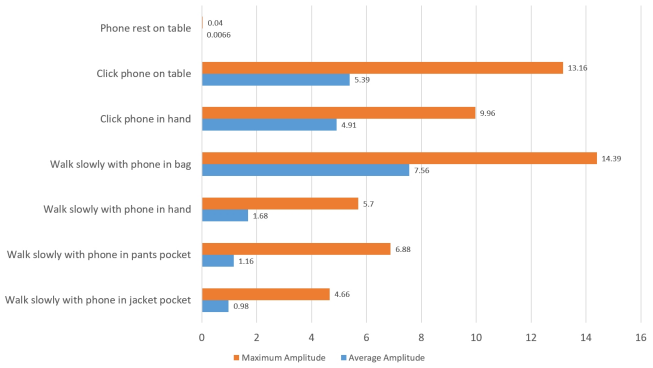



Fig. 6: Avg. vs. Max. amplitude in 7 scenarios.

classify this stationary state into two distinct categories: a completely stationary state and an audio playback state. The completely stationary state is the best attack scenario. In the completely stationary state, users do not interact with their phones for an extended period, and their phones do not engaged in any activities that could influence sensor readings. Conversely, in the audio playback state, it is assumed that users put their phones aside to watch videos or listen to musics, so they will not pick up their phones.

To further test the impact of playing audio on VibLeak, we played music while transmitting files, set the volume to 50% and 100%, and measured the decibel value at a distance of 20cm from the phone. The results are shown in Table IV. In the scenario involving 50% volume, both the LRA and BRA exhibited a decrease, while the accuracy experienced a more noticeable decline in the 100% volume scenario. After analyzing the data, we found that when the volume is 100%, the amplitude value of the vibration generated by the music has exceeded which generated by the motor, so the music will affect VibLeak. However, when the volume is 50%, the amplitude of the vibration generated by the music is small and will not affect VibLeak.

TABLE IV: Phone status impact

| Phone Scene | LRA | BRA | dB |
|---|---|---|---|
| Rest, playing music at 50% volume | 90% | 93.36% | 64 |
| Rest, playing music at 100% volume | 80% | 90.71% | 77 |

### F. Transmission Environment Experiment

In this subsection, we focus on the impact of different material surfaces on VibLeak. We specifically examined two surface types encountered in everyday life: soft surfaces and hard surfaces, to more accurately reflect real-world usage scenarios. Soft surfaces are characterized by their ability to absorb a portion of vibrational energy. To simulate this scenario, we conducted experiments on a bed. Subsequently, we investigated hard surfaces, which facilitate more direct vibration propagation but may also produce increased echoes and resonances. We chose wooden and steel table as the hard surface experimental environment. To simulate the real environment, we additionally divided each different surface

environment into scenarios with and without a phone case. The experimental equipment and data are consistent with Section III-C.

TABLE V: Transmission environment impact

| Environment | BRA | LRA |
|---|---|---|
| Bed | 97.78% | 95% |
| Wooden table | 99.78% | 100% |
| Iron plate | 99.50% | 95% |

Table V presents the findings from our experiments. The comparative analysis of the results shows that when the mobile phone is placed on a soft surface (such as a bed), the soft surface will absorb some vibrations, and the transmission effect is poorer compared to the hard surface environment. However, due to the absorption of some vibrations on soft surfaces, the phone produces less sound when vibrating on soft surfaces.

### G. Vibration Concealment

In addition to examining strategies to enhance the transmission success rate, it is essential to address the aspect of concealment during the actual transmission process. We conducted a series of tests using various combinations of vibration intensity, measuring decibel levels at a distance of 20 centimeters from the vibrating device. We divide concealment into two levels based on decibel levels in quiet environments: quietness(26∼31dB) and slight noise(31∼36dB). The experimental equipment and data are the same as in Section III-C, and results are illustrated in Figure 7.

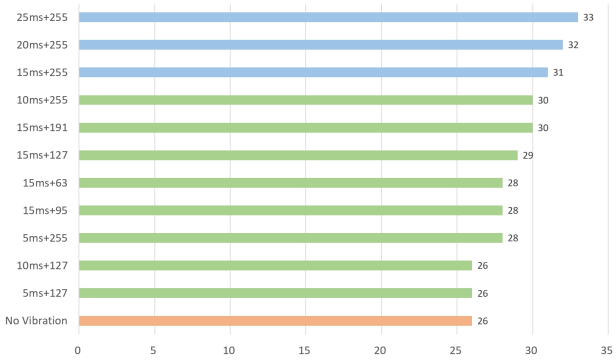



Fig. 7: dB levels under different intensity combinations.

The vertical axis in the figure represents the grouping of vibration duration and amplitude, while the horizontal axis represents decibel values. Among them, the orange line represents the ambient sound level without vibration, and the corresponding decibel value belongs to normal quietness, serving as the reference group for the experiment. The green lines represent a normal quietness level similar to the baseline group, which requires conscious close listening to distinguish the presence of vibrations. The blue lines represent the vibration combination belonging to the second level, with a slight increase in sound compared to the reference group. People can hear the vibration sound at close range in a quiet environment.

## IV. Conclusion

In this paper, we propose an innovative covert-channel attack called VibLeak, which utilizes smartphone motors and motion sensors to convert private files into vibration signals, which are then transmitted discreetly. We evaluate the effectiveness and stealthiness of VibLeak in various smartphones and real-world scenarios.


## Acknowledgement

The work of Junye Jiang, Jingcheng Ju, Zhenlu Tan and Duohe Ma in this paper was supported by National Natural Science Foundation of China(No.62472418).


## References


[1] B. Mishra, A. Agarwal, A. Goel, A. A. Ansari, P. Gaur, D. Singh, and H.-N. Lee, "Privacy protection framework for android," *IEEE Access*, 2022.

[2] J. Yu, L. Lu, Y. Chen, Y. Zhu, and L. Kong, "An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing," *IEEE TMC*, 2021.

[3] L. Cai and H. Chen, "Touchlogger: inferring keystrokes on touch screen from smartphone motion," USENIX Association, 2011.

[4] 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," *ArXiv*, vol. abs/1602.04115, 2016.

[5] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *USENIX Security 14*, 2014.

[6] M. Gao, Y. Liu, Y. Chen, Y. Li, Z. Ba, X. Xu, J. Han, and K. Ren, "Device-independent smartphone eavesdropping jointly using accelerometer and gyroscope," *IEEE TDSC*, 2023.

[7] S. A. Anand and N. Saxena, "Speechless: Analyzing the threat to speech privacy from smartphone motion sensors," in *IEEE SP*, 2018.

[8] A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen, "Spearphone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers," *ArXiv*, vol. abs/1907.05972, 2019.

[9] Z. Ba, T. Zheng, X. Zhang, Z. Qin, B. Li, X. Liu, and K. Ren, "Learning-based practical smartphone eavesdropping with built-in accelerometer," *NDSS*, 2020.

[10] Y. Liang, Y. Qin, Q. Li, X. Yan, L. Huangfu, S. Samtani, B. Guo, and Z. Yu, "An escalated eavesdropping attack on mobile devices via low-resolution vibration signals," *IEEE TDSC*, 2023.

[11] I. Hwang, J. Cho, and S. Oh, "Privacy-aware communication for smartphones using vibration," *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2012.

[12] T. Yonezawa, J. Nakazawa, and H. Tokuda, "Vinteraction: Vibration-based information transfer for smart devices," in *IEEE ICMU*, 2015.

[13] N. Roy, M. Gowda, and R. R. Choudhury, "Ripple: Communicating through physical vibration," in *NSDI*, 2015.

[14] N. Roy and R. R. Choudhury, "Ripple {II}: Faster communication through physical vibration," in *NSDI*, 2016.

[15] K. Lee, V. Raghunathan, A. Raghunathan, and Y. Kim, "Syncvibe: Fast and secure device pairing through physical vibration on commodity smartphones," in *IEEE ICCD*, 2018.

[16] S. Sen and D. Kotz, "Vibering: Using vibrations from a smart ring as an out-of-band channel for sharing secret keys," in *ICIOT*, 2020.

[17] R. Wijewickrama, S. Dohadwalla, A. Maiti, M. Jadliwala, and S. Narain, "Skinsense: Efficient vibration-based communications over human body using motion sensors," *Internet Things*, 2023.

[18] K. Cui, Q. Yang, Y. Zheng, and J. Han, "mmripple: Communicating with mmwave radars through smartphone vibration," in *IPSN*, 2023.