Understanding Side-Channel Eavesdropping Attack

Chunyu Xia
cxia@ucsd.edu

Tianyi Liu
t8liu@ucsd.edu

***Abstract*—With the prevalence of Voice User Interfaces, the related privacy issues became a popular topic, especially the potential threat for side-channel eavesdropping. We roughly divided the side-channel eavesdropping attacks into three main tasks, including Automatic speech recognition (ASR), Spoken Language understanding (SLU) and Command Classification. In this report, we tried to understand side-channel eavesdropping attack from sampling rate and frequency response. First of all, in order to investigate the importance of sampling rate, we resampled the audio signal and test them in our pretrained model, and realized that sampling rate is the key factor for speech-based tasks due to the correpsonding number of data points. Moreover, we measured the frequency response and noise model from motion sensors in the smartphone, and tested the mimicked data with three tasks. From the results, we proved that motion sensor side-channel eavesdropping is more suitable for SLU and command classification with limited categories, and perform badly for ASR tasks.**

***Index Terms*—Side-channel eavesdropping, Automatic speech recognition, Spoken Language understanding, Command Classification, sampling rate, frequency response**

I. Introduction

Speech has been the most wildly used wireless communication method far early before the invention of wireless network. Eavesdropping is to secretly steal private information without users' awareness, so eavesdropping attacks for speech or conversation has always been a big concern. In addition, with the development of Artificial Intelligence (AI) and pervasiveness of mobile devices, Voice Control Systems (VCS) like Voice Assistant (VA) have become a popular research and application field. In the meantime, related eavesdropping issues have also become more and more severe since eavesdropping attacks expose threats for privacy leakage between human and machine.

Microphone is the common tools utilized to steal private conversation, but there are several defenses or protection methods. Nonetheless, side-channel eavesdropping attack is a new threat domain where little work has been done. The basic idea of side-channel eavesdropping is that adversaries use channels that are not aware by the users to capture the side effects of sound waves like vibration, which is generated by sound sources or caused by nearby sound reflector. Due to the limitation of hardware design, such attacks are hard to be prevented. Even though such side-channels are not designed as microphone to record users' audio, the vibration of loudspeakers also contain huge privacy information especially when users have conversation with voice assistant.

Loudspeaker and human throat are two main types of sound sources that can emit human sound, and are two main channel to be attacked by related side-channel eavesdropping attack. For example, loudspeaker is vulnerable to side-channels like motion sensors and in-built hardwares which can log the vibration from loudspeakers. Human throat is threatened by high-resolution laser or millimeter wave radar because there will be movement of human throat when human speaks. What's more, the location and the type of sound source will also change the side-channel eavesdropping attack. In this paper, we only focus on the loudspeaker related side-channel eavesdropping attack, because side-channel like motion sensors are more feasible to be measured and motion sensors based side-channel eavesdropping is becoming prevalent in recent years.

Loudspeakers are easily attacked by in-built sensors, which can be called internal eavesdropping, while human throat is threatened by sensors outside, which can be called external eavesdropping. In this report, we only focus on in-built sensors eavesdropping like motion sensors, and we assume that the two key factors that make side-channel signals heavily different from audio signals are sampling rate and frequency response.

Most eavesdropping attacks can be categorized into three different speech tasks: ASR, SLU and Command Classification. Although some side-channel eavesdropping attack papers mention their incredible results for eavesdropping tasks, such eavesdropping attacks are mostly constrained in specific experiment settings and open to explore more insights about the real-world scenarios.

The goal of this paper is to generate the motion sensor side-channel eavesdropping data and check its threat for different tasks with various sampling rate. Our contribution in this work is shown as follows.

- Investigate how sampling rate will influence the eavesdropping related models
- Measure the frequency response and noise model from motion sensors in the smartphone
- Mimic the motion sensors data from audio signals with sampling rate, frequency response and noise
- Test the generated motion sensors data for our previous tasks

The rest of the paper is organized as follow: Section II discusses some related work for internal and external eavesdropping. Section III lists the datasets, metrics and tasks for side-channel eavesdropping attack. Section IV shows the importance of sampling rate. Section V explains the detail about measuring the motion sensor and tests such side-channel modality. Section VI is the conclusion of our report with some analysis and future plans.

II. Related Work

Side-channel Eavesdropping attack for speech is a well-known privacy threat. Internal Eavesdropping means that the attack is dependent on the in-built sensor or hardwares to reconstruct the sound signals. Motion sensors are the ideal target to be utilized for attack because motion sensors are known as zero-permission on mainstream mobile OS like Android and iOS. Recent work uses accelerometers [1] [2] [3] [4], gyroscopes [5] to record the vibration generated by loudspeaker. Moreover, previous study [4] shows that such attack can be achieved only when the sound source and the motion sensors are set on the same surface. The key difficulty of motion sensor based speech eavesdropping is the low sampling rate (< 500Hz), so previous work focus more on simple words, touchstone or command instead of speech. Other internal eavesdropping attack needs additional devices or modified hardwares. VibraPhone [6] utilizes the modified vibra motor as "microphone". HDH [7] shows the possibility that the magnetic hard disk drives can be used as microphones by capturing the acoustic wave's oscillations.

External eavesdropping attack means that the sound source vibration is recorded by external sensors like RF signals and laser/light. RF signals, like WiFi, UHF RFID, UWB and mmWve signals, can be regarded as external measurement for sound source vibration. ART [8] uses WiFi signals to capture the signal strength and phase change caused by loudspeaker vibrations. UWHear [9] is trying to use Impluse Radio Ultra-Wideband to sense the vibration. Moreover, mmWave Radar has been shown huge threat for side-channel eavesdropping. Waveear [10] shows the possibility to develop a direct eavesdropping for human throat because of the short wave length and high-resolution of mmWave Radar. Laser/light based external eavesdropping is also attractive. It is possible to utilize a laser beam to detect sound vibrations in a distant object. Lamphone [11] tries to use the vibration of hanging light bulb as side-channel to recover the speech.

III. Datasets, Metrics and Tasks

The goal of this section is to demonstrate the feasibility of speech recognition by reconstructing the voice from data captured by the motion sensor in real time. In this part, we have divided our effort into three tasks. For each of our three tasks, we evaluated the recognition rate and accuracy using distinct datasets and measurement variables. As a baseline, we used different read English speeches at a sampling rate of 16 kHz and resampled them down. We're interested in seeing how the result changes as the sample rate goes down.

A. Datasets

The first dataset we used is **Librispeech**. LibriSpeech is a corpus which contains approximately 1000 hours of read English speech, which is collected by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from read audiobooks from the LibriVox project. [12]Additionally, it includes the pretrained language model that we used in our first task.

The second data set we used is **Google Speech Command** dataset. The Google Speech Command dataset, as its name implies, is provided by Google, containing 65,000 one-second long utterances of 30 short words, recorded by thousands of different people, contributed by public through the AIY website. [13] Compared to other speech data sets available at the time, the GSC dataset is regarded as less complex. It is small and efficient for simpler tasks.

The third data set we used is **Fluent Speech Commands**. The Fluent Speech Commands dataset contains 97 speakers saying 30,043 utterances. In total, the dataset contains 248 phrases that correspond to 31 distinct intents that are divided into three categories: action, object, and location. [14] The pre-trained machine learning models using this dataset can directly extract the intent from speech, rather than having to convert it to text before doing so.

The last dataset we used is **Timers and Such**. Time and Such is a collection of spoken English commands intended for use with common voice control systems. In contrast to the Google Speech Command, this dataset uses cases involving numbers. The dataset has four intents, corresponding to four common offline voice assistant uses: SetTimer, SetAlarm, SimpleMath, and UnitConversion. The semantic label for each utterance is a dictionary with the intent and a number of slots. [15]

B. Metrics

The Word Error Rate (WER) is for assessing the accuracy of automatic speech recognition (ASR) systems in converting voice to text. It is a useful tool for comparing different systems as well as evaluating improvements within one system.

It can be calculated using the following formula:

$$WER = \frac{S + I + D}{N}$$

Where, **S** stands for substitutions (replacing a word). **I** stands for insertions (inserting a word). **D** stands for deletions (omitting a word). **N** is the number of words that were actually said. [16]

Bilingual Evaluation Understudy (BLEU) is an algorithm used to evaluate the quality of machine translation of natural language words and sentences. The core concept is that the quality of translation depends on the corresponding relationship between the output of machine translation and human translation. The closer the machine translation is to the professional human translation result, the better the performance is.

The improved evaluation metric, unigram's precision, the most widely used metric in BLEU, can be calculated as:

$$P = \frac{min(w_{max}, w_{test})}{m}$$

where

- **m** is the total number of words in the test phrase.
- w_{max} is the maximum total count of that word in the candidate translations.
- w_{test} is the count of that word in the test phrase.

[17]

If we have more than one reference, for each word, BLEU first determines the maximum matching word count among all references as w_{max}. It then chooses the minimum of that value between the test phrase word count w_{test} and w_{max}. Finally, it divides the min value with the total words in test phrase.

The Character Error Rate (CER) is another widely used metric for assessing the accuracy of automatic speech recognition (ASR) systems. However, unlike the WER, which calculates the number of words that did not interpret correctly, the CER calculates the number of characters.

The CER can be calculated using the following formula:

$$CER = \frac{S + I + D}{N}$$

Where each variable is the same with WER, except they are on a scale of character not a word.

Classification Accuracy is the most common metric used to evaluate the performance of a classification predictive model. Accuracy may be calculated in a straightforward manner using the formula:

$$Accuracy = CorrectPredictions/TotalPredictions.$$

The Sentence Error Rate (SER), like the CER and WER, is another widely used metric for assessing the accuracy of ASR systems in converting voice to text. SER calculates the number of sentences in a speech that is incorrect.

The SER can be calculated using the following formula:

$$SER = \frac{S + I + D}{N}$$

Where each variable is the same with WER, except they are on a scale of sentence not a word.If a single character in a sentence is different, then the whole sentence will be classified as a wrong translation. So the SER will be larger than that of WER or CER.

C. Tasks

The Automatic Speech Recognition (ASR) system was the first task we worked on. Automatic Speech Recognition (ASR), alternatively referred to as speech-to-text or machine speech recognition, is a technique through which a machine converts a speech signal to the corresponding text or command after recognizing and understanding it. When transcribing the spoken speech into written text, it is common to use a machine learning model that minimizes the Word Error Rate (WER) metrics as much as possible.

The second task we worked with is called Command Classification. This task required us to categorize the commands into a discrete set of categories. For example, some classes can be directions (up, down, left, right...) or commands such as open and close. Our model makes use of the Key Word Spotting algorithm. With this algorithm, a model can constantly analyze speech patterns in order to detect certain "command" classes.

The third task we did is Spoken Language Understanding (SLU). The term has largely been coined for targeted understanding of human speech directed at machines. While the first task, automatic speech recognition (ASR), attempts to convert a speaker's spoken utterances into text strings, SLU attempts to deduce the intentions of the user from their speech utterances. [16]

IV. Resampling

Human sound ranges from 30Hz to 10kHz, and most of the energy is concentrated from 300 to 3400Hz. According to Nyquist Sampling Theorem, in order to record full band of the speech signals, audio signals at least have 16kHz sampling rate, so the most of audio signals range from 16kHz to 48kHz. However, the sampling rate of side-channel attack is pretty low due to the hardware limitations. In this part, we will demonstrate the results for different speech related tasks with various sampling rate to show the importance of sampling rate in side-channel attack.

First of all, we divide the whole range of sampling rate into three parts, including Low sampling rate (0 to 1000Hz), medium sampling rate (1000 to 4000Hz) and high sampling rate (> 4000Hz).

- Low sampling rate (0-1000Hz): The common motion sensors in the smartphones have the sampling rate lower than 500Hz. Even though the maximum sampling rates of the motion sensor chips can support is around 4000Hz, the sampling rate constrained by smartphone OS is less than 500Hz. Mainstream Android smartphones like SamSung S21, OnePlus 7 Pro all support sampling rate higher than 400Hz, but iPhone only has limited sampling rate around 100Hz. Moreover, commercial WiFi router can achieve 500-1000Hz sampling based on stable Channel State Information (CSI). Low sampling rate modality is still possible to recognize digital number and key words.
- Medium sampling rate (1000-4000Hz): UWB radar, high speed video camera and WiFi signals can achieve 1000-2000Hz sampling rate, so these modalities show the opportunity to recover the speech in the experiment setting.
- High sampling rate (> 4000Hz): High sampling rate should be achieved by self-designed ADC, and it is close to the original microphone sampling rate. It is possible to recover arbitrary speech by the existing pretrained speech recognition model.

Sampling rate is the fundamental factor for side-channel eavesdropping attack. Even though existing papers demonstrate that side-channels with low sampling rate can be used to recognize the specific words in a limited dictionary and side-channels with high sampling rate is able to recover the speech. In this part, we will show the effect of sampling rate for different tasks.

We simply use resampled audio files to simulate different sensors, and we use evaluation metrics developed based on these data sets to evaluate recognition rates at various resampled rates. According to the Librispeech test results (Table I), there is a noticeable increase in both WER and CER, as well as a decrease in BLEU precision between 2000 Hz and 1000 Hz.

	16000	8000	4000	2000	1000	500	250	125	50
WER	0.28%	1.14%	3.13%	13.35%	86.93%	96.59%	100.00%	100.00%	100.00%
BLEU	99.40%	97.84%	92.78%	73.99%	16.48%	5.14%	0.00%	0.00%	0.00%
CER	0.14%	0.43%	1.37%	8.61%	61.02%	82.50%	99.34%	98.96%	98.68%

TABLE I
RESULT FOR LIBRISPEECH

	16000	8000	4000	2000	1000	500	250	125	50
ACC	98.00%	96.50%	88.30%	52.10%	19.50%	10.50%	2.20%	0.10%	0.00%

TABLE II
RESULT FOR GOOGLE SPEECH COMMAND

	16000	8000	4000	2000	1000	500	250	125	50
1-SER	94.00%	88.00%	54.00%	2.00%	2.00%	0.00%	0.00%	0.00%	0.00%

TABLE III
RESULT FOR FLUENT SPEECH COMMANDS

	16000	8000	4000	2000	1000	500	250	125	50
1-SER	100.00%	100.00%	82.00%	22.00%	8.00%	8.00%	6.00%	6.00%	6.00%

TABLE IV
RESULT FOR TIMERS AND SUCH

We discovered an intriguing result when we examined the results from Google Speech Command (Table II). GSC's accuracy is twice that of Librispeech (1-WER) at 500Hz. GSC maintains a 2.2 percent accuracy at 250 Hz, while Librispeech's accuracy drops to zero at the same rate. The sampling rate of our phone's motion sensors is 500Hz. This result indicates that, while the motion sensor theoretically does not interfere with the ASR task, it is still capable of recognizing words or commands.

For the SLU task result (in Table III), the accuracy decreased to 2% at the sampling rate of 2000Hz for the fluent speech command data, which indicates that simply lowering the sampling rate would let the command audio lack of information to be categorize without interpreting to text. This task's outcome indicates that complex commands are invulnerable to attacks at low sampling rates. However, the timers and such result (in Table IV) demonstrated an incredible accuracy of 22% at 2000Hz and even 8% at 500Hz. This indicates the possibility of a side channel attack eavesdropping on numerical information via motion sensors.

V. FREQUENCY RESPONSE

From the definition from Wikipedia, frequency response is the quantitative measure of the output spectrum of a system or device in response to a stimulus, and is used to characterize the dynamics of the system. The frequency responses are different for various side-channels and do not have the flat frequency response for the speech bandwidth. Internal eavesdropping attack is easy to measure while external eavesdropping attack is hard to measure because of the multipath effect and the propagation channels from different placements. In this part, we will measure the frequency response of motion sensors in smartphone, and try to mimic the motion sensors only with audio signals.

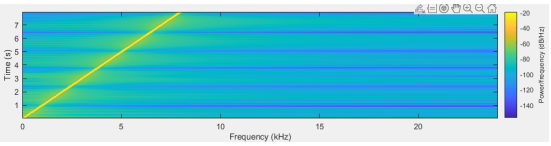

Fig. 1. 8s FMCW ranging from 0 to 8000Hz

Figure 1 shows our well-designed Frequency-Modulated Continuous Wave (FMCW) ranging from 0 to 8000Hz in 8 seconds, and we choose the sampling rate 48kHz for FMCW audio signals. We directly measure the frequency response by choosing the maximum value in the spectrogram.

We play such FMCW audios in the smartphone to measure the frequency response of motion sensor (accelerometer). Figure 2 shows the plot of accelerometer signals with played FMCW, and Figure 3 shows the spectrogram. We can find that there are waves shown in the spectrogram due to the low sampling rate of the accelerometer in the smartphone.

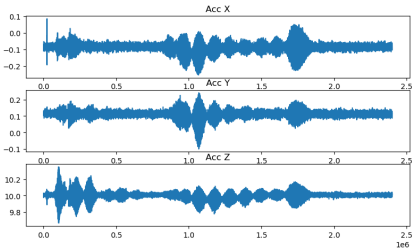

Fig. 2. Plot of Accelerometer with FMCW

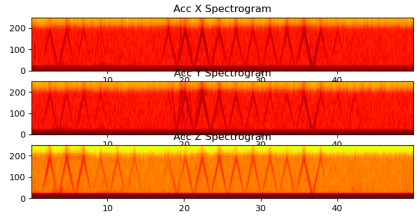

Fig. 3. Spectrogram of Accelerometer with FMCW

Next, we upsample the accelerometer signals to 48kHz, which is the same as the FMCW audio signals. Figure 4 shows the cross correlation result with 250Hz FMCW, so we can find the beginning of the measurement signals.

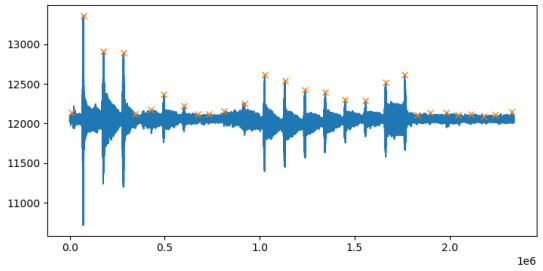

Fig. 4. Cross-correlation Result

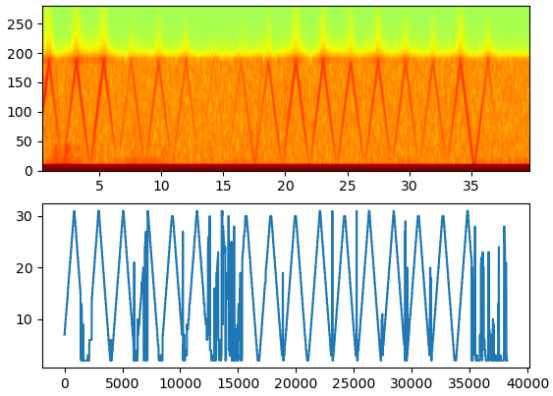

Fig. 5. Spectrogram and frequency mapping

We generate the spectrogram after synchronization, and select the maximum index of the spectrogram as frequency mapping as shown in the Figure 5.

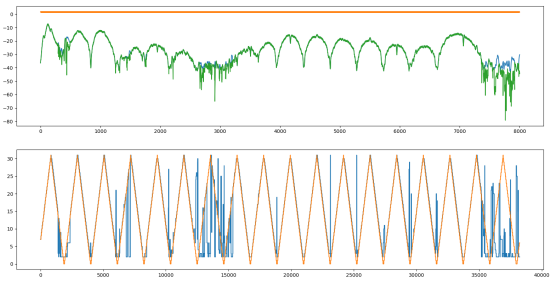

Fig. 6. Frequency Response of Accelerometer

Figure 6 shows the result for accelerometer frequency response. Green line shows the frequency response based on the fitting slope, blue line shows the frequency response based on the maximum value, and yellow line shows the frequency response based on the transmitted signals. The yellow line show the groundtruth, and the generated FMCW audio has the flat frequency response. We discover that blue and green line is close to each other, and we choose the green line as our result.

After successfully measuring the frequency response for accelerometer in smartphone, we design a finite impulse response (FIR) filter to fit the frequency response. Figure 7 show the result for FIR filter, and we set filter order to 10000 to better fit the frequency response.

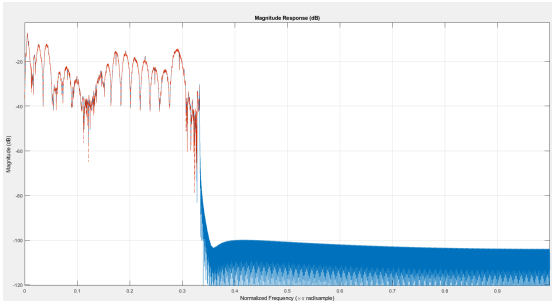

Fig. 7. FIR filter to fit the frequency response of accelerometer

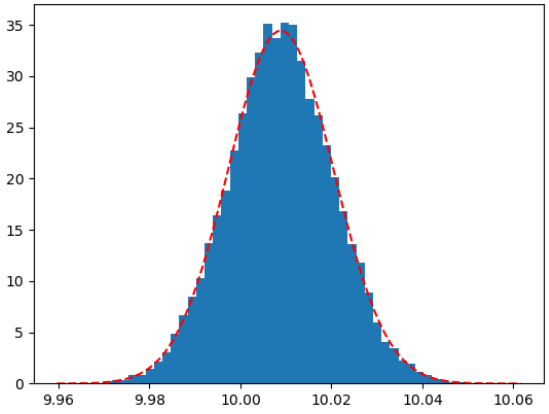

Fig. 8. Normal distribution to fit the noise

We then use the normal distribution to the noise from accelerometer. Figure 8 shows the result normal distribution with $\mu = 10, \sigma = 0.01$.

Eventually, we add the Gaussian noise and frequency response to the audio signals. Figure 9 represents the generated accelerometer signals only from the audio signals, and Figure 10 is the groundtruth. Our generated signals have high cross-correlation value with the original signals as shown in Figure 11, so we assume that our artificial data successfully mimic the accelerometer data. We will this pipeline to test motion sensor side-channel eavesdropping for different tasks.

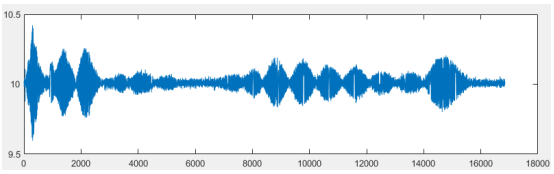

Fig. 9. Generated accelerometer signals from audio

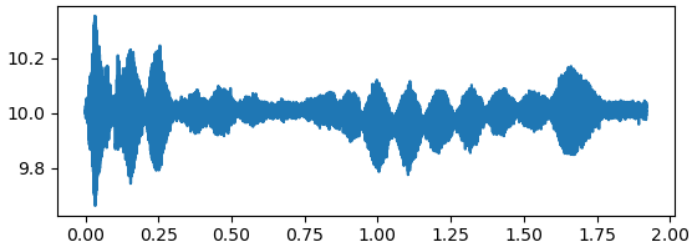

Fig. 10. Original accelerometer signals

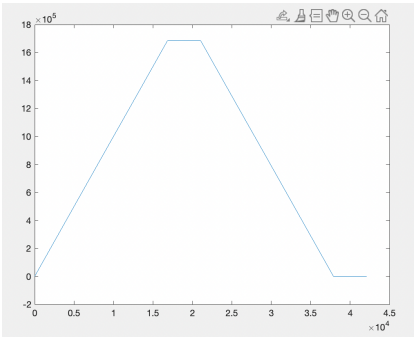

Fig. 11. Cross-Correlation between original and generated signals

For ASR task in Table V, even though the sampling rate affects the performance badly, the performance in the beginning is not stable. We dig into the recognition result, and discover that the recognition result makes no sense even for the original sampling rate. The generated side-channel data is beyond the scope of the pretrained model, so the WER, CER and BLEU are pretty bad and the recognition result is just random guess. This finding proves that accelerometer side-channel attack is not suitable for ASR tasks, specifically for large vocabulary.

For command classification task in Table VI, the accuracy is only 54.27% even for the original sampling rate, but recognizing the command with 2000Hz sampling rate is still possible because of the 24.97% accuracy, which means accelerometer data is vulnerable to command classification problem with few categories.

For SLU task in Table VII, we find that it maintains 12% accuracy for accelerometer even with 500Hz sampling rate, which shows the potential real-world threat for our fluent speech. Table VIII demonstrates the bad performance in timers-and-such dataset.

VI. Conclusion

In this report, we first analyze the effect of sampling rate in different tasks, and realize that sampling rate is the fundamental factors for speech-related tasks. The main reason behind is that the model is hard to make decision for limited data points if we have low sampling rate. Moreover, we discover that 1000-2000Hz is the watershed between high and low recognition rate for ASR task, and the result for pretrained model with low sampling rate ($< 1000Hz$) is pretty bad. For command classification and SLU tasks, the model can still get some true prediction with 1000Hz, but the model fails with even lower sampling rate, which again proves the previous study that data with low sampling rate is suitable for easier tasks like command classification and SLU task.

For accelerometer side-channel attack, we use frequency response and fitted noise model to mimic the accelerometer data only with audio signals. Our cross-correlation results show the high similarity between original and generated signals. We use the generated signals for different tasks mentioned above. We find that such side-channel attack is not suitable for existing ASR pretrained model even with high sampling rate, but it demonstrates the potential threat for command classification and SLU tasks. Even though the recognition performance is lower with the generated data, our pretrained models can still predict the true answers with 2000Hz sampling rate. Moreover, for Fluent Speech Commands dataset, our model still has 12% accuracy with 500Hz sampling rate. In summary, the motion sensor side-channel attack is a real-world threat for eavesdropping tasks, especially for the speech tasks with limited categories.

In the future, we will measure more side-channel modalities including Vibra motor, HDD, WiFi, RFID and mmWave Radar, and pretrain the models by ourselves with these side-channel modalities instead of using the existing pretrained models. We will try to understand the side-channel eavesdropping attacks with measurement work.

References

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

[2] S. A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen, "Spearphone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers," *arXiv preprint arXiv:1907.05972*, 2019.

[3] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohapatra, "Accelword: Energy efficient hotword detection through accelerometer," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 2015, pp. 301–315.

[4] S. A. Anand and N. Saxena, "Speechless: Analyzing the threat to speech privacy from smartphone motion sensors," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 1000–1017.

[5] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, 2014, pp. 1053–1067.

[6] N. Roy and R. Roy Choudhury, "Listening through a vibration motor," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 57–69.

[7] A. Kwong, W. Xu, and K. Fu, "Hard drive of hearing: Disks that eavesdrop with a synthesized microphone," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 905–919.

[8] T. Wei, S. Wang, A. Zhou, and X. Zhang, "Acoustic eavesdropping through wireless vibrometry," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 130–141.

	16000	8000	4000	2000	1000	500	250	125	50
WER	88.60%	84.97%	85.49%	85.49%	90.67%	97.41%	100.00%	100.00%	100.00%
BLEU	8.94%	12.26%	13.22%	13.35%	9.12%	1.20%	0.00%	0.00%	0.00%
CER	69.56%	66.33%	64.11%	60.24%	71.96%	88.38%	99.54%	99.26%	97.32%

TABLE V
SIDE-CHANNEL RESULT FOR LIBRISPEECH

	16000	8000	4000	2000	1000	500	250	125	50
ACC	54.27%	46.20%	37.33%	24.97%	1.90%	0.60%	0.10%	0.00%	0.00%

TABLE VI
SIDE-CHANNEL RESULT FOR GOOGLE SPEECH COMMAND

	16000	8000	4000	2000	1000	500	250	125	50
1-SER	86.00%	54.00%	24.00%	12.00%	8.00%	12.00%	6.00%	6.00%	6.00%

TABLE VII
SIDE-CHANNEL RESULT FOR FLUENT SPEECH COMMANDS

	16000	8000	4000	2000	1000	500	250	125	50
1-SER	26.00%	24.00%	10.00%	8.00%	0.00%	0.00%	0.00%	0.00%	0.00%

TABLE VIII
SIDE-CHANNEL RESULT FOR TIMERS AND SUCH

[9] Z. Wang, Z. Chen, A. D. Singh, L. Garcia, J. Luo, and M. B. Srivastava, "Uwhear: through-wall extraction and separation of audio vibrations using wireless signals," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 1–14.

[10] C. Xu, Z. Li, H. Zhang, A. S. Rathore, H. Li, C. Song, K. Wang, and W. Xu, "Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, 2019, pp. 14–26.

[11] B. Nassi, Y. Pirutin, A. Shamir, Y. Elovici, and B. Zadov, "Lamphone: Real-time passive sound recovery from light bulb vibrations," *Cryptology ePrint Archive*, 2020.

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," 2015.

[13] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[14] L. Lugosch, M. Ravanelli, P. Ignoto, V. Singh Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *Interspeech 2019*, 2019.

[15] L. Lugosch, P. Papreja, M. Ravanelli, A. Heba, and T. Parcollet, "Timers and such: A practical benchmark for spoken language understanding with numbers," *NeurIPS 2021*, 2021.

[16] M. Gevirtz, "What is word error rate (wer)?" Sep 2021. [Online]. Available: https://deepgram.com/blog/what-is-word-error-rate/

[17] K. Wołk and K. Marasek, "Enhanced bilingual evaluation understudy," *arXiv preprint arXiv:1509.09088*, 2015.