2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS) | 979-8-3315-1596-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICPADS63350.2024.00013

BreathPass: Ultrasounic Authentication by Chest and Abdomen Movement while Breathing

Lingkun Li, Fan Dang†, Duo Liu, Zhichao Cao‡
Beijing Jiaotong University, †Tsinghua University, ‡Michigan State University

***Abstract*—In this study, we propose BreathPass, a non-invasive authentication system that characterizes the chest/abdomen movement incurred by human breath to enable unlocking smart devices while wearing various types of face covers, clothing, in different postures, and dynamic status such as walking or running. To capture the breathing pattern, BreathPass uses speakers to emit ultrasound signals. The signals are reflected off the chest wall and abdomen and then back to the microphone, which records the reflected signals. The system then extracts the breathing pattern from the reflected signals, and further extracts fingerprints from the breathing pattern, and use these fingerprints to perform authentication. We carefully design a Deep Neural Network model and explore its capacity for feature abstraction in order to address the challenges associated with tiny position changes resulting in different breathing patterns and the extremely narrow bandwidth of breathing. We implement a prototype and conduct extensive experiments. BreathPass achieves an overall accuracy of 83%, a true positive rate of 73%, and a false positive rate of 5%, according to performance evaluation results.**

I. Introduction

With the advancement of modern smart devices, unlocking methods have shifted away from the "what you know" schema and toward the "who you are" schema. With the "what you know" method, a user needs to pre-configure some information such as PINs and secret questions, and the device will then challenge the user to verify that she or he actually owns the device. Such a PIN is often complex to ensure security and makes it difficult for individuals to remember to some level. In addition, these passcodes or answers are vulnerable facing blindly replay-attack since the devices do not care who is entering the information. With "who you are" tactics, the user no longer needs to type in the complex PIN, thus simplifying and speeding up unlocking. These approaches are quite popular with users because of their non-invasive nature and ease of use; *e.g.,* Apple employs Face-ID to unlock the iPhone and iPad via facial recognition [1]. Apart from facial recognition, fingerprint identification is a frequently used method for unlocking smart gadgets [2]. In addition, voiceprint recognition [3], iris recognition [4], heartbeat recognition [5], breathing voice recognition [6], gaze gesture [7], and tooth-edge recognition [8] also plays a key role in biometric recognition approaches.

These approaches, however, have drawbacks in two different aspects.

Vulnerable to Replay-attack: Some of them are still compromised by replay-attacks, *e.g.,* many research efforts [9], [10], [11], [12], [13] focus on resolving the replay-attack among voiceprint-based, fingerprint-based, gaze-based, or face-based authentication. For example, we could spoof others' face and voice with masks and recording.

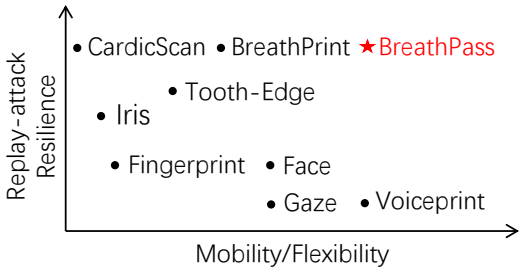


Fig. 1. Comparasion of existing biometric authentication methods.

Lack of Mobility and Flexibility: Other approaches using iris, tooth-edge, heartbeat, and human breath are not sufficiently flexible on mobile devices, *e.g.,* iris-based authentication requires the device to equip specific designed components such as inferred cameras, meanwhile, it needs users to look at a specific area to make sure that the inferred camera could capture a clear iris. Heartbeat-based authentication such as Cardiac Scan [5] requires the deployment of two radar sensors, which are not standard hardware and so have a high operating cost. BreathPrint [6] is a novel approach that doesn't need to equip specific designed components and can significantly defend against replay attacks, however, it cannot work in some scenarios including some people choose to wear a mask as the breathing voice that is needed by the system could be blocked by the face cover. The face cover also makes Smileauth [8] infeasible since it requires an image of the tooth-edge which is blocked by the face mask.

In this paper, we propose BreathPass, a new non-invasive breath-based "who you are" authentication technique. BreathPass detects users' breath in a non-invasive manner, extracts features from their breath, and then verifies that the user is permitted. As shown in Figure 1, BreathPass is a novel approach that is hard to be compromised by replay attacks because breath pattern is hard to be spoofed and imitated. In addition, BreathPass is flexible enough since it only uses commercial off-the-shelf (COTS) components equipped on almost every devices, and can be used in a wider scenarios such as wearing different kinds of face covers and clothes, in different postures, and in different dynamic status such as walking or running. BreathPass faces the following challenges in order to implement it and achieve all of the aforementioned

979-8-3315-1596-6/24/$31.00 ©2024 IEEE
DOI 10.1109/ICPADS63350.2024.00013

requirements:

1) As with BreathPrint, face covers may obstruct the voice of the user's breath. To overcome this challenge, BreathPass should avoid using a microphone to record users' breathing voices; instead, we employ an ultrasound-based chest wall and abdomen motion-sensing technology to characterize users' breathing patterns. Specifically, BreathPass works by initially emitting ultrasonic waves through the speaker of a smart device, such as a smartphone. The ultrasonic waves then travel to the user's chest wall and abdomen, where they are reflected back to the smart device's microphone. The motion of the chest wall and abdomen, which characterizes human breath, alters the phase of the reflected signal, and such phase shifts are used to authenticate;

2) Unlike the speaker verification [14], [15] which normally converts the speech signal to a spectrogram in order to extract features, the motion of the chest wall and abdomen typically has an extremely low frequency of less than 1 Hz. As a result, features derived from spectrograms such as *Mel Frequency Cepstral Coefficients* (MFCC) or *Gammatone Frequency Cepstral Coefficients* (GFCC) [16] cannot be used to identify the breath. To address this issue, we implement the authentication mechanism using a one-dimensional Convolutional and Siamese Neural Network. Specifically, the neural network takes two raw chest wall and abdominal motion waveforms as the input. One of these two inputs is the template input collected from the enrollment stage, while the other is the matching input collected from the authentication stage. During training the neural network, it learns the breathing pattern and generates a vector of features, saying fingerprint, which can be used to calculate the distance between two inputs. Finally, BreathPass uses the distance between the two inputs to determine if they originate from the same person or not;

3) Unlike mechanical vibration, which typically has a stable frequency [17], breathing patterns between individuals do not share the same prior knowledge as mechanical vibration. Additionally, even when people are in the same posture, their breathing patterns may vary. In other words, small movements result in different breathing patterns. As a result, denoising the motion of the chest wall and abdomen requires developing a model that can suppress the moving-dependent noise while retaining the user-dependent difference. To address this issue, we introduce a technique called average fingerprinting. With such a technique, the template input is composed of multiple chest wall and abdomen motion signals that might come from different tiny postures. BreathPass generates multiple fingerprints from template signals using a neural network. Following that, the system computes the average of those fingerprints and then uses that average fingerprint to determine the distance to the fingerprint obtained during the authentication stage. Finally, it calculates the authentication result using that distance.

Our contributions are listed as follows:

- We design a novel mechanism for sensing human breathing patterns and build a DNN to determine whether the breathing pattern provided by the user is authorized.
- By using the breath sensing mechanism and the DNN we built, we create BreathPass, which enables smart devices to perform authentication via the human breath. We also implement a proof-of-concept software to evaluate BreathPass's performance.
- On the basis of our implementation, we conduct extensive experiments. BreathPass achieves an 83% accuracy, a 73% true-positive rate (TPR), and a 5% false-positive rate (FPR) in general, according to the experiment results. The BreathPass system is stable when the user is wearing a variety of different face covers, clothing, and postures. We believe that in the future, it may be a candidate for a "who you are" unlocking mechanism, or it may serve as a complement to other untrustworthy mechanisms, such as eye recognition, in order to provide authentication services jointly.

II. Design

A. Ultrasound-based Breath Sampler

We use the speakers on smart devices, such as a smartphone, to play a stereo ultrasound signal. The speaker is perpendicular to and close to the chest wall. The left channel plays an ultrasound signal at an 18 kHz frequency, while the right channel plays one at a 22 kHz frequency. The ultrasound signals are reflected off the chest wall and abdomen and are picked up by the smartphone's microphone, which is also positioned near the chest wall. Formally, the signal emitted is denoted as follows:

$$s(t) = \cos(2\pi f_1 t) + \cos(2\pi f_2 t), \tag{1}$$

where $f_1 = 18,000$ and $f_2 = 22,000$. After the microphone records the reflected signal $m(t)$, the breath sampler first employs a high pass filter to eliminate components below 16 kHz. Then, inspired by the previous efforts [18], [19], the breathing pattern can be regarded as the signal $x(t)$ modulated into $m(t)$ by the carrier of $s(t)$. Therefore, we have

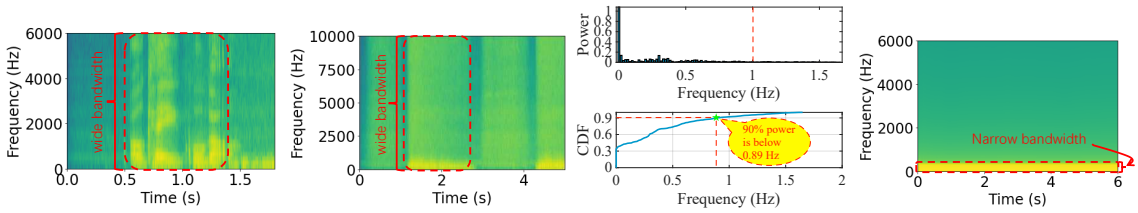
$$m(t) = x(t)s(t). \tag{2}$$

To demodulate the breathing pattern $x(t)$, we need to multiply $m(t)$ by $s(t)$ and let the result pass through a low pass filter with an extremely low cutoff frequency, *e.g.,* 200 Hz. From Equation (1) and (2), we have

$$\begin{aligned} m(t)s(t) = x(t)s^2(t) &= x(t)[\cos(2\pi f_1 t) + \cos(2\pi f_2 t)]^2 \\ &= x(t)[\cos^2(2\pi f_1 t) + 2\cos(2\pi f 1_t)\cos(2\pi f_2 t) \\ &\quad + \cos^2(2\pi f_2 t)] \\ &= x(t)\{\frac{1}{2}[1+\cos(2\pi 2 f_1 t)] + \cos(2\pi(f_1+f_2)t) \\ &\quad + \cos(2\pi(f_2-f_1)t) + \frac{1}{2}[1+\cos(2\pi 2 f_2 t)]\}. \end{aligned} \tag{3}$$

After a low pass filter with a 200 Hz cutoff frequency, the components $\cos(2\pi 2 f_1 t)$, $\cos(2\pi 2 f_2 t)$, $\cos(2\pi(f_1+f_2)t)$, and $\cos(2\pi(f_1-f_2)t)$ all disappear. Therefore, we have

$$m(t)s(t) \Longrightarrow x(t)(\frac{1}{2}+\frac{1}{2}) = x(t). \tag{4}$$



(a) Speech spectrogram (b) Breathe sound spectrogram (c) FFT of a breathing pattern (d) Breathing pattern spectrogram

Fig. 2. (a) Spectrogram of a speech "OK, Google!". (b) Spectrogram of a breathing sound. (c) FFT and CDF of a breathing pattern. (d) Spectrogram of a breathing pattern

We use the extracted $x(t)$ as the breathing pattern to perform authentication.

B. Fingerprint Extractor Design

Design Issues: After sampling the breathing pattern, both the enrollment stage and the authentication stage all send their samples to the fingerprint extractor. In order to get a feasible fingerprint that can be used to perform authentication, the design of fingerprint extractor should take the following challenges into consideration:

1) *Denoise.* Previous works [17], [19], [20] proposed multiple approaches to denoise the vibration waveform or the breathing waveform for machine damage or human disease detection. For example, mmVib [17] reports the machine error when an abnormal vibration is detected. The system collects the vibration waveform with noises and leverages a model to denoise the signal. After that, the system will measure the distance between the sampled signal and the normal status signal. If the distance is within a threshold, then the system classifies the machine works normally. Otherwise, the system reports the machine in an abnormal status. To build such a denoise model, the system usually collects vibration waveform with noises when the machine works normally and use a series of transformation and processes, *e.g.,* matching arc on the I-Q plane [17], to match the noise waveform with the standard vibration waveform as precise as possible. After matching, it fixes the processes and parameters to denoise future signals. If the machine works in abnormal status, the signal after being processed by the same model with the same parameters is far from the standard one. Such an idea was also adopted by SpiroSonic [19] and BreathListener [20] to detect if human breathes normally. The common point of these works is to find the identical pattern when the machine or the lung works normally. In BreathPass, however, the goal is to characterize the difference in the breathing pattern among different people instead of finding the typical pattern from different peoples' breaths. Therefore, it is hard for us to build a denoise model by extracting the common pattern.

2) *Stability.* The chest wall and the abdomen motion are not as stable as a machine. A different breathing pattern may be extracted even the user stays in the same posture, but after a tiny movement; *e.g.,* when a user leans back on a chair from the straight waist, a different breathing pattern will be extracted. Therefore, the design of the fingerprint extractor must take such a stability issue into consideration.

3) *Feature selection.* The most similar task to BreathPass is speaker verification. The speaker verification task first uses a microphone to record the user who is saying a predefined sentence or any other sentence. The system then verifies if the recorded voice comes from the authorized users. Existing speaker verification solutions typically first transform the voice signal into the spectrogram, then extract spectrogram-based features such as *Mel Frequency Cepstral Coefficients* (MFCC) or *Gammatone Frequency Cepstral Coefficients* (GFCC) [16]. After that, such solutions leverages the *Gaussian Mixture Model* (GMM) or build a Deep Neural Network (DNN)-based model to verify the speaker.

BreathPrint [6] adopts a similar idea. Different from the speaker verification task is that it uses a microphone to record the user's breathing voice, then extracts GFCC features and leverages the GMM model to verify if the breathing voice comes from the authorized user.

To extract spectrogram-based features, the system needs to get a signal with a reasonable wide bandwidth; *e.g.,* speaker verification and BreathPrint [6] are both capable of leveraging spectrogram-based features since the spectrogram of a speech "OK, Google!" could reach up to 6 kHz as shown in Figure 2(a), and the spectrogram of a breath sound can reach up to 10 kHz as shown in Figure 2(b). Therefore, there is a sufficient area to embed information in the spectrogram so that these systems are able to use photo verification-liked models to perform authentication.

In BreathPass, however, we cannot use spectrogram-based features since the bandwidth of a breathing pattern is extremely narrow. An adult typically finishes a breathing cycle in 2 to 3 seconds. We plot the result of the Fast Fourier Transform (FFT) of a breathing pattern that we sampled with the method described in Section II-A as shown in Figure 2(c). We can find that the majority of the power is under 1 Hz (90% of power is below 0.89 Hz). Therefore, the bandwidth of a breathing pattern in the spectrogram is too narrow as shown in Figure 2(d) to provide sufficient information that can be used to perform authentication.

Our Solution: To address the first challenge that we cannot build a denoise model based on observation, instead, we build a DNN-based model to learn how to denoise the signal and extract the fingerprint itself. To cope with the third challenge, we use the raw breathing pattern waveform as the input instead of extracting spectrogram-based features.

As shown in Figure 3, the fingerprint extractor is consists

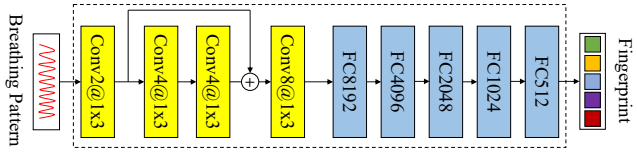


Fig. 3. The structure of our DNN model for fingerprint extractor.

of a series of convolutional layers followed by some fully connected layers. Each layer uses the ReLU function to activate, and there is a max-pooling layer with a length of 4 after the first and the last convolutional layers. Each fully connected layer except the last one adopts dropout with the parameter 0.2 to avoid overfitting. We also adopt the idea of ResNet [21] of adding a skip link between convolutional layers to avoid gradient vanishing. The fingerprint extractor takes a breathing pattern waveform sampled by the method described in Section II-A as the input. After the last fully connected layer of the extractor, it outputs a vector of 512 floating-point numbers as the fingerprint.

The second challenge about stability requires us to remove moving-dependent noises while reserving the user-dependent difference among different users' breathing patterns. To cope with this issue, we introduce the average fingerprint technique. Specifically, instead of using the fingerprint that comes from a single breathing pattern waveform as the result of the enrollment stage, we sample multiple breathing patterns in the enrollment stage and get multiple fingerprints correspondingly. We calculate the average of these fingerprints as the result of the enrollment stage.

The idea behind the average fingerprint is that if we focus on the same user, moving-dependent noises are unstable while the user-dependent difference is stable, therefore, if we take the average of multiple fingerprints, unstable moving-dependent noises will be smoothed while the stable user-dependent difference is reserved. We show the effectiveness of the average fingerprint in Section IV-I.

C. Comparator Design

After getting fingerprints from both the enrollment stage and the authentication stage, we need to build a comparator to measure the distance between two fingerprints.

After getting the fingerprints, we build the comparator by applying logistic regression. Specifically, we have the target function

$$f(x, y) = \sigma(w^T \|x - y\|_2 + b), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function, w is the vector of parameters of the comparator, b is the bias, and x and y are the fingerprints from the enrollment stage and the authentication stage, respectively. During training the comparator, the target output $f(x, y)$ is set to 1 if the x and y are from the same user, otherwise, the target output is set to 0.

D. Combine the Fingerprint Extractor with the Comparator

As shown in Figure 4, we put these components together. The fingerprint extractor and comparator are combined into

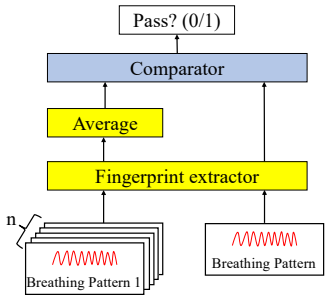


Fig. 4. The end-to-end system design combining the fingerprint extractor with the comparator

a single neural network. During the training process, we randomly choose n breathing patterns from the same volunteer in the training dataset and choose another breathing pattern from a random volunteer in the training dataset. If these two volunteers are the same one, then the final result, *i.e.,* `Pass?`, is set to 1; otherwise, it is set to 0.

As for the deployment of these components, the left lower side of the figure, *i.e.,* n breathing patterns, comes from the enrollment stage, and we store the average fingerprint in advance after the enrollment stage. During the authentication stage, the system samples a breathing pattern as shown on the right lower side of the Figure 4. If the output of the comparator is greater than 0.5, we denote the final result, *i.e.,* `Pass?`, as 1, indicating that authentication was successful; otherwise, we denote the final result as 0, indicating that authentication failed. If authentication fails, BreathPass prompts the user to sample his breathing pattern and attempt authentication again; if authentication continues to fail, BreathPass will prevent the user from sampling his breathing pattern until the user enters the correct PIN number.

III. Implementation

A. Breathing Pattern Sampler and Data Collection

We develop the breathing pattern sampler on Android smartphones. We use the native Android library *AAudio* [22] to generate, emit, and record the ultrasound waves. We use our sampler to collect data and extract the breathing patterns of 20 volunteers. Each volunteer is continuously sampled for 60 seconds and five times (i.e., 300s in total). The 20 volunteers cover people of different gender and age that may frequently use smart devices. We use a Google Pixel 3a smartphone running Android 11 to perform sampling. The sampling rate is set as 48 kHz. During sampling, we place the smartphone on a desk and let the speaker towards a volunteer's chest. The distance between the smartphone and the volunteer is between 5 and 10 cm. An interval separates two consecutive samplings to allow the volunteer to adjust their tiny posture. Once the microphone samples the reflected ultrasound signals, we leverage *Apache Commons Math* package [23] to build a high pass filter that eliminates all components below 16 kHz, leaving only the ultrasound signals and then extracts the breathing pattern using the design described in Section II-A.

B. Training the Feature Extractor and Comparator

After getting the dataset from 20 volunteers, we build the feature extractor and comparator using PyTorch on a desktop equipped with an NVIDIA GeForce RTX 3090 GPU, as discussed in Section II-B and Section II-C. We randomly select 10 volunteers' data for the training set and the remaining volunteers' data for the test set. During each iteration of the training and testing, we first randomly select a volunteer, then we randomly choose a 60s long breathing pattern from the indexed volunteer dataset, and finally, we randomly crop a segment of the 60s long breathing pattern ranging from 1s to 5s. This process is repeated 10 times to create the template inputs. Then we get another segment of breathing pattern but alternatively choose the same volunteer and a different volunteer as the authentication input. If we have chosen the same volunteer, the target of the DNN output is set to 1; otherwise, we set it to 0. We also add some fake breathing patterns which are collected by the breathing pattern sampler with the smartphone speaker towards the wall or towards nothing to enhance the classification accuracy. We always set the target of the DNN output to 0 if any of these fake patterns are chosen.

IV. Evaluation

A. Overview

To evaluate BreathPass, we use the data we collected to train and test BreathPass. In general, we use the following metrics to evaluate the performance of BreathPass:

Accuracy: We use accuracy to determine whether BreathPass can correctly identify the authorized user whose fingerprints are stored during the enrollment stage. The accuracy is calculated as

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} I(\hat{y}_i = y_i)}{N}, \tag{6}$$

where N is number of test cases, I is the indicator function, $\hat{y}_i$ is the output given by BreathPass, and y_i is the correct label. In general, the greater the accuracy, the better.

True positive rates (TPR) and false positive rates (FPR): Besides the accuracy, we also focus on two metrics, *i.e.,* true positive rates (TPR) and false positive rates (FPR). We calculate the TPR by using the equation

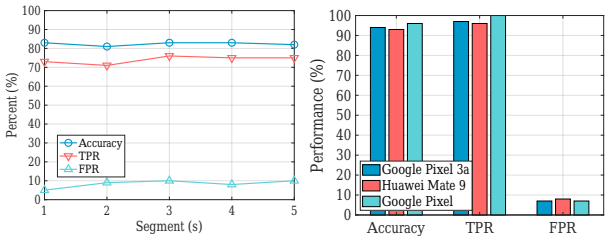
$$\text{TPR} = \frac{\sum_{i=1}^{N} I(\hat{y}_i = 1 \text{ and } y_i = 1)}{\sum_{i=1}^{N} I(y_i = 1)}, \tag{7}$$

and the FPR is calculated by

$$\text{FPR} = \frac{\sum_{i=1}^{N} I(\hat{y}_i = 1 \text{ and } y_i = 0)}{\sum_{i=1}^{N} I(y_i = 0)}, \tag{8}$$

where N is the number of test cases, I is the indicator function, $\hat{y}_i$ is the output given by BreathPass, and y_i is the correct label.

When an enrolled user attempts to unlock the device, the TPR determines the likelihood that the system will successfully authenticate. When an unauthorized user attempts to unlock the device, the FPR determines the possibility that the system will pass the authentication by mistake. The higher the TPR, the better, while the lower the FPR, the better.



(a) (b)

Fig. 5. (a) General performance of BreathPass (b) Performance of different mobile models

Apart from the TPR and FPR, two additional metrics are used to characterize the authentication system's performance: true negative rates (TNR) and false negative rates (FNR), which indicate the likelihood of an unauthorized user being successfully blocked by the system and the likelihood of an authorized user failing the authentication, respectively. However, we are unconcerned with these two values because an attacker cannot do anything if the device cannot be unlocked.

B. General Evaluation

Setup: To determine whether the extracted breathing pattern can be used for authentication, we train and test the fingerprint extractor and comparator using the dataset we collected. We have formed the training dataset by randomly choosing 10 volunteers from the whole dataset. In this experiment, we use the remaining 10 volunteers as well as the fake breathing patterns (generated from motions other than breathing) to perform testing. We perform 1000 iterations of testing. During each iteration, we randomly choose 1s to 5s breathing pattern segments to form the test datasets with mini-batches of 32, resulting in a total of $32 \times 1000 = 32000$ test cases. The output of the sigmoid function in Equation (5) is in the range of [0, 1]. If the output is greater than or equal to 0.5, the result is considered passed; otherwise, the result is considered failed.

Results: As shown in Figure 5(a), BreathPass achieves over 80% accuracies, over 70% TPRs, and less than 10% FPRs for any segment length of the input breathing pattern. BreathPass achieves an accuracy of 83%, a TPR of 73%, and an FPR of 5% when the input breathing pattern is segmented for 1 second, which is the best segment length. As a result, we assert that the breathing pattern we sampled can be used for authentication and that when combined with the TPR and the FPR, BreathPass can serve as a candidate for a "who you are" scheme.

C. Effectiveness on Different Mobile Models

Setup: To verify whether BreathPass is able to work on different mobile models, we launch BreathPass on three different mobile phones, *i.e.,* Google Pixel 3a, Huawei Mate 9, and Google Pixel. In this experiment, we use these three mobile models to collect a volunteer's breathing pattern respectively. Then we form the positive test cases by selecting pairs of 1s

breathing patterns from the collected breathing pattern. After that, we make pairs of the breathing patterns from a given mobile model and the test datasets as discussed in Section IV-B as the negative cases.

Results: As shown in Figure 5(b), there is little difference between accuracies, TPRs, and FPRs among different mobile models. Therefore, BreathPass can work on different mobile models.

D. Influence of Different Kinds of Face Covers

Wearing a face cover may obstruct airflow into the user's nose or mouth, thereby altering the user's breathing pattern. To characterize the effect of various types of face covers on users, we prepared four types of commonly used face covers, *i.e.,* surgical, fabric, KN95, and N95, as shown in Figure 6(a). There are almost no blocks when wearing the surgical mask, while the remaining makes breathing harder than wearing the surgical mask or not wearing a face cover. We would like to characterize the performance across different kinds of face covers. In this experiment we only care about TPRs, which means the possibility of successfully authenticated while wearing different face covers.

Setup without grouping: In this experiment, we invite a volunteer to wear each of the four types of face covers separately and evaluate the BreathPass's performance. We ask the volunteer to enroll his breathing pattern with no face cover, and perform authentication with wearing different kinds of face covers.

Results without grouping: As shown in figure 6(b), TPRs decrease while wearing the masks which blocks the airflow, but almost all of them are over 40%, which means that the extracted breathing pattern is still feasible while wearing different kinds of face covers.

Setup with grouping: To further improve the TPRs while wearing the face covers that blocks the airflow, we can split the face covers into two groups, *i.e.,* no airflow blocked (no face covers and surgical) and airflow blocked (fabric, KN95 and N95). We ask the volunteer to enroll with one of them in a group and perform authentication with wearing another one in that group. Specifically, we firstly use breathing patterns collected without a face cover to generate the template fingerprint and use breathing patterns collected with the surgical mask as the input of the authentication stage. Then we use the KN95 dataset to generate the template fingerprint and use breathing patterns from the fabric, and the N95 dataset, respectively, as the input of the authentication stage. Finally, we generate the template fingerprint using breathing patterns from the N95 dataset and use it to evaluate performance when wearing the KN95 mask. This is reasonable, as the user could enroll in both groups separately and choose one manually or automatically before performing the authentication.

Results with grouping: As shown in figure 6(c), compare to the TPRs without face cover, all TPRs with a face cover are decreased, but most of the TPRs are higher than 70%, and in particular, for 1s, the TPRs are all higher than 80%. Therefore, BreathPass is feasible across different face covers.

E. Influence of Different Clothes

Setup: Since BreathPass extracts breathing patterns from the motion of the chest wall and the abdomen, the breathing pattern collected might be influenced by different clothes because different wearings might have different effects of blocking ultrasound signals. In this experiment, we choose the most common used four kinds of clothes, *i.e.,* T-shirt, hoodie, sweater, and jacket, as shown in Figure 6(d), and invite a volunteer to sample his breathing patterns while wearing these clothes, correspondingly. We then use the dataset collected with wearing the T-shirt to generate the template fingerprint, and use the breathing pattern from datasets with wearing all four clothes correspondingly as the input of the authentication stage.

Results: As shown in figure 6(e), the TPRs are almost higher than 65%, and in particular, for 1s, the TPRs are all over 75%. Therefore, BreathPass is feasible across different kinds of clothes.

F. Influence of Different Postures

Setup: As discussed in Section II-A, different postures result in different breathing patterns. Therefore, to characterize the influence of different postures, we invite a volunteer to provide breathing patterns with the three most common postures, *i.e.,* sitting, standing, and laying down. We use breathing patterns extracted from sitting posture to generate the template fingerprint, and use breathing patterns extracted from all these three postures respectively as inputs of the authentication stage.

Results: As shown in figure 7(a), the sitting and standing posture have higher TPRs than laying down. The TPRs for sitting and standing are almost higher than 60%, and in particular, for 1s, the TPRs are all over 70%. Therefore, BreathPass is feasible across different postures.

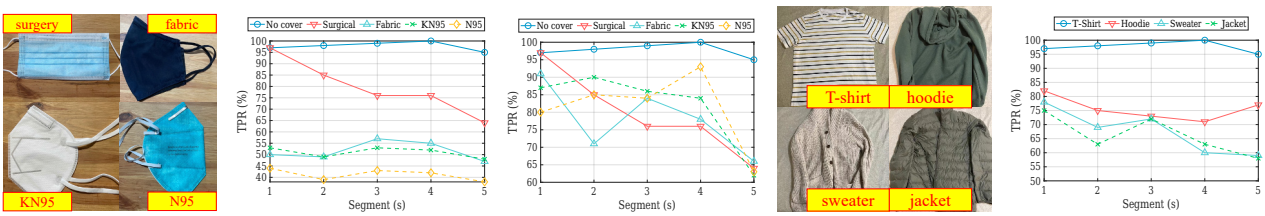
G. Influence of Dynamic Status

Setup: Some dynamic status such as walking or after running may result in different breathing pattern, to verify if BreathPass could still successfully authenticate the user under these dynamic status. We ask a volunteer to enroll his breathing pattern while sitting in a quiet room at rest, and perform authentication while sitting in a quiet room at rest (marked baseline), during walking, and after running 500m, respectively. We choose 1s as the segment length because, from the previous experiments, we find that 1s segment length works well for most cases.

Results: As shown in table I, walking has almost no effect to authentication. Authentication after running has a bigger effect as it significantly changes the breathing pattern, however, it still achieves 78% of the TPR, which means that BreathPass is feasible when the user is under dynamic status.

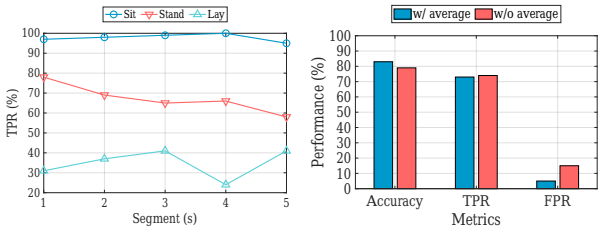
H. Influence of Different Environments

Setup: To verify if BreathPass could still successfully authenticate under different environments. We ask a volunteer to enroll his breathing pattern while sitting in a quiet room



(a) Masks (b) TPR w/o grouping (c) TPR w/ grouping (d) Clothes (e) TPR

Fig. 6. Performance of BreathPass with different kinds of masks and clothes



(a) Posture TPR (b) Performance w/ or w/o average

Fig. 7. (a) TPR of BreathPass with different postures. (b) Performance with or without average fingerprint technique.

TABLE I
TPRS OF DIFFERENT DYNAMIC STATUS AND ENVIRONMENTS.

Class	Baseline	Walking	Running	Outside	TV
TPR	97%	94%	78%	78%	80%

at rest, and perform authentication while sitting in a quiet room at rest (marked baseline), outside while raining (lower noise), and near a TV set playing a concert with a high volume (higher noise), respectively. We choose 1s as the segment length because, from the previous experiments, we find that 1s segment length works well for most cases.

Results: As shown in table I, compare to baseline, authentication outside while raining and near the TV set decrease the TPR. It probably because the raining falling down between the speaker and the chest wall affects the transmition and reflection of the ultrasound signals, and suppression effects of the microphone [24] affects the recording quality when background noise is huge. The TPRs however, are around 80%, which means that BreathPass is feasible under different environments.

I. Effectiveness of the Average Fingerprint

Setup: During our experiment, we found that even all volunteers in Section III-A are sitting while sampling their breathing patterns, the DNN-based model also cannot get a good performance. This is because even a tiny move within the same posture could result in different breathing patterns that affect the overall performance. As discussed in Section II-B, we introduce an average fingerprint technique to eliminate moving-dependent noises while reserving user-dependent differences. In this experiment, we build another model of the same DNN architecture as discussed in Section II but without the average fingerprint technique. We choose 1s as the segment length because, from the previous experiments, we find that 1s segment length works well for most cases. We compare the performance between models with and without the average fingerprint technique to show the effectiveness of the average fingerprint technique. Specifically, we use the same training dataset to train the same model without the average fingerprint technique. After the model converges, we use the same test dataset as Section IV-B to test the performance.

Results: As shown Figure 7(b), we can find that the accuracy without the average technique is lower than the model with the average technique. We can further find that although they have close TPRs, the FPR without the average technique is much higher than the model with the average technique, which is unacceptable. The reason why the model without the average technique has a high FPR is because the model cannot eliminate moving-dependent noises, thus taking them as the feature to construct the fingerprint. Therefore, it is necessary to apply the average fingerprint technique so that the model could successfully eliminate moving-dependent noises while reserving user-dependent differences.

J. Efficiency on Mobile Phones

Setup: To make BreathPass practical, the DNN-model needs to finish the inference on a mobile device within a reasonable time limit after a user samples his breathing pattern. To test the efficiency of BreathPass, we port our model on a Google Pixel 3a Android mobile phone. The application shows the time used by the DNN-model along with the authentication results. We perform 10 times of authentication. The configuration is the same as the previous experiments, and we use 1s segment length of breathing patterns as the inputs. Specifically, we enroll 10 breathing signals that each of them is 1s long, and extract 10 fingerprints, respectively, and store them on the smartphone. During the authentication stage, after the user samples his 1s breathing pattern, we first calculate the average of 10 fingerprints, then take the result of the average and sampled breathing pattern as the input to the model. The model extracts the fingerprint of the sampled breathing pattern and runs the comparator to give the result of the authentication.

Results: We calculate the average running time, and the result is 855.7 ms, which shows that BreathPass can be used practically.

V. RELATED WORKS

Ultrasound sensing systems can be used to complete a variety of sophisticated tasks. For example, existing research efforts [25], [26] employ ultrasound signals to detect sleep

apnea. Specifically, these works emit modulated ultrasound, *i.e.,* FMCW chirp or pseudo-white noise signal, and then use a classification algorithm to determine whether an apnea symptom exists. Moreover, SpiroSonic [19] uses reflected ultrasonic signals to detect whether the user's pulmonary function is normal. BreathListener [20] also uses reflected ultrasonic signals to quantify the driver's breathing status, thereby determining whether or not the driver is driving safely. AcuTe [27] measures ambient temperature via ultrasonic sensing by utilizing the linear relationship between temperature and sound speed.

VI. Conclusion

In this paper, we propose BreathPass, a novel biometric authentication method that is more resilience to replay-attack and has a high flexibility to mobile devices. It samples breathing patterns from users and extracts fingerprints from them to achieve authentication. We believe that BreathPass can become a candidate of "who you are" unlocking mechanism, or become complementary to another untrustable mechanism such as eye recognition to provide authentication service together.

Acknowledgement

This work is supported in part by the Talent Fund of Beijing Jiaotong University (No. 2022XKRC013), Natural Science Foundation of China under Grant No. 62402028, 62302259, 62072029, Beijing Nova Program (20230484263), and Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Qilu University of Technology (Shandong Academy of Sciences) under Grant No. 2023ZD039.

References

[1] About Face ID advanced technology - Apple Support. https://support.apple.com/en-us/HT208108. (Accessed on Nov. 04, 2021).

[2] In-screen fingerprint sensors coming to 100 million phones by 2019? - cnet. https://www.cnet.com/tech/mobile/in-screen-fingerprint-sensors-coming-to-100-million-phones-by-2019-report/. (Accessed on Nov. 04, 2021).

[3] Zia Saquib, Nirmala Salam, Rekha Nair, and Nipun Pandey. Voiceprint recognition systems for remote authentication-a survey. *International Journal of Hybrid Information Technology*, 4(2):79–97, 2011.

[4] John Daugman. New methods in iris recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5):1167–1175, 2007.

[5] Feng Lin, Chen Song, Yan Zhuang, Wenyao Xu, Changzhi Li, and Kui Ren. Cardiac scan: A non-contact and continuous heart-based user authentication system. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 315–328, 2017.

[6] Jagmohan Chauhan, Yining Hu, Suranga Seneviratne, Archan Misra, Aruna Seneviratne, and Youngki Lee. Breathprint: Breathing acoustics-based user authentication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 278–291, 2017.

[7] Yinghui Li, Zhichao Cao, and Jiliang Wang. Gazture: Design and implementation of a gaze based gesture control system on tablets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–17, 2017.

[8] Hongbo Jiang, Hangcheng Cao, Daibo Liu, Jie Xiong, and Zhichao Cao. Smileauth: Using dental edge biometrics for user authentication on smartphones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–24, 2020.

[9] Yongchao Ye, Lingjie Lao, Diqun Yan, and Lang Lin. Detection of replay attack based on normalized constant q cepstral feature. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 407–411. IEEE, 2019.

[10] S Saranya, Suvidha Rupesh Kumar, and B Bharathi. Deep learning approach: detection of replay attack in asv systems. In *International Conference on Soft Computing and Signal Processing*, pages 291–298. Springer, 2019.

[11] Bin Hao, Xiali Hei, Yazhou Tu, Xiaojiang Du, and Jie Wu. Voiceprint-based access control for wireless insulin pump systems. In *2018 IEEE 15th international conference on mobile ad hoc and sensor systems (MASS)*, pages 245–253. IEEE, 2018.

[12] Miroslav Goljan, Jessica Fridrich, and Mo Chen. Defending against fingerprint-copy attack in sensor-based camera identification. *IEEE Transactions on Information Forensics and Security*, 6(1):227–236, 2010.

[13] Roberto Caldelli, Irene Amerini, and Andrea Novi. An analysis on attacker actions in fingerprint-copy attack in source camera identification. In *2011 IEEE International Workshop on Information Forensics and Security*, pages 1–6. IEEE, 2011.

[14] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE, 2016.

[15] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[16] Xiaojia Zhao, Yang Shao, and DeLiang Wang. Casa-based robust speaker identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(5):1608–1616, 2012.

[17] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. mmvib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.

[18] Wei Wang, Alex X Liu, and Ke Sun. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 82–94, 2016.

[19] Xingzhe Song, Boyuan Yang, Ge Yang, Ruirong Chen, Erick Forno, Wei Chen, and Wei Gao. Spirosonic: monitoring human lung function via acoustic sensing on commodity smartphones. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[20] Xiangyu Xu, Jiadi Yu, Yingying Chen, Yanmin Zhu, Linghe Kong, and Minglu Li. Breathlistener: Fine-grained breathing monitoring in driving environments utilizing acoustic signals. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 54–66, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[22] Android. Aaudio library. https://developer.android.com/ndk/guides/audio/aaudio/aaudio. (Accessed on Nov. 02, 2021).

[23] Apache. Commons math: The apache commons mathematics library. https://commons.apache.org/proper/commons-math/. (Accessed on Nov. 02, 2021).

[24] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 2–14, 2017.

[25] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. Contactless sleep apnea detection on smartphones. In *Proceedings of the 13th annual international conference on mobile systems, applications, and services*, pages 45–57, 2015.

[26] Anran Wang, Jacob E Sunshine, and Shyamnath Gollakota. Contactless infant monitoring using white noise. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[27] Chao Cai, Zhe Chen, Henglin Pu, Liyuan Ye, Menglan Hu, and Jun Luo. Acute: acoustic thermometer empowered by a single smartphone. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 28–41, 2020.