MicPro: Microphone-based Voice Privacy Protection

Shilin Xiao
USSLab, Zhejiang University
Hangzhou, China
xshilin@zju.edu.cn

Xiaoyu Ji*
USSLab, Zhejiang University
Hangzhou, China
xji@zju.edu.cn

Chen Yan
USSLab, Zhejiang University
Hangzhou, China
yanchen@zju.edu.cn

Zhicong Zheng
USSLab, Zhejiang University
Hangzhou, China
zheng_zhicong@zju.edu.cn

Wenyuan Xu
USSLab, Zhejiang University
Hangzhou, China
wyxu@zju.edu.cn

ABSTRACT

Hundreds of hours of audios are recorded and transmitted over the Internet for voice interactions such as virtual calls or speech recognitions. As these recordings are uploaded, embedded biometric information, i.e., voiceprints, is unnecessarily exposed. This paper proposes the first privacy-enhanced microphone module (i.e., MicPro) that can produce anonymous audio recordings with biometric information suppressed while preserving speech quality for human perception or linguistic content for speech recognition. Limited by the hardware capabilities of microphone modules, previous works that modify recording at the software level are inapplicable. To achieve anonymity in this scenario, MicPro transforms formants, which are distinct for each person due to the unique physiological structure of the vocal organs, and formant transformations are done by modifying the linear spectrum frequencies (LSFs) provided by a popular codec (i.e., CELP) in low-latency communications.

To strike a balance between anonymity and usability, we use a multi-objective genetic algorithm (NSGA-II) to optimize the transformation coefficients. We implement MicPro on an off-the-shelf microphone module and evaluate the performance of MicPro on several ASV systems, ASR systems, corpora, and in real-world setup. Our experiments show that for the state-of-the-art ASV systems, MicPro outperforms existing software-based strategies that utilize signal processing (SP) techniques, achieving an EER that is 5 ∼ 10% higher and MMR that is 20% higher than existing works while maintaining a comparable level of usability.

CCS CONCEPTS

• **Security and privacy → Privacy protections**; **Usability in security and privacy**.

KEYWORDS

voiceprint protection; anonymization; microphone; CELP codec

*Xiaoyu Ji is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CCS '23, November 26–30, 2023, Copenhagen, Denmark.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0050-7/23/11…$15.00
https://doi.org/10.1145/3576915.3616616

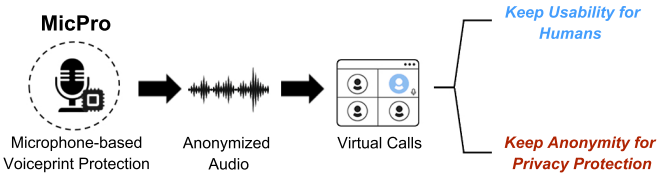


Figure 1: MicPro illustration: MicPro protects the privacy of users at the microphone module level, and it can produce an anonymous recording with suppressed biometric information for applications such as real-time virtual calls.

ACM Reference Format:
Shilin Xiao, Xiaoyu Ji, Chen Yan, Zhicong Zheng, and Wenyuan Xu. 2023. **MicPro: Microphone-based Voice Privacy Protection. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3576915.3616616**

1 INTRODUCTION

The advances in speech recognition and sensing technologies have enabled a boost in voice-controllable systems, speech-to-text devices, and voice assistant services. It has been reported that hundreds of hours of voices are recorded and uploaded to the cloud per minute [4] by smart speakers, smartphones, video and audio conference platforms, etc. Since a voice recording contains not only speech information but also voice biometric information that can identify a human being (aka. voiceprints), these voice data impose privacy risks where attackers can gain access or the voice technology companies themselves may misuse it: One can illegally extract victims' voiceprint to impersonate them by bypassing authentication [55, 60] or perform identity inference to gain sensitive information [30, 31].

In this paper, we aim to design anonymization methods that target low-latency scenarios such as virtual calls. We propose privacy-by-design paradigms called MicPro[1], i.e., privacy-enhanced microphone modules that will produce anonymous audio recordings without jeopardizing the usability, as shown in Fig. 1. The recordings shall preserve linguistic content and sound naturally for a human being if they ever need to be heard.

[1]https://github.com/USSLab/MicPro

Previous anonymization schemes face two challenges in these scenarios: (1) most of them are incompatible with the low-latency, frame-by-frame transmissions because they require a full audio piece instead of a frame as the input. (2) The anonymization process becomes futile if an attacker can obtain the original audio before it is anonymized at the software level. For example, an attacker might be able to compromise the victim's device and gain control of the microphone or steal local audio files directly [22, 23]. To address these issues, `MicPro` works at the microphone module level to provide low-latency, low-overhead anonymization. Once `MicPro` produces the audio recordings, they can be stored locally or in the cloud, and they will protect users' voiceprints.

To the best of our knowledge, this is the first work to develop microphone modules that can suppress identifiable biometric information contained within recordings of speech while preserving speech quality. The idea sounds promising yet challenging. First of all, `MicPro` should not require modifying the hardware of existing microphone modules and have to produce anonymous recordings in low latency with limited computational capability and embedded functionality. Prior works that rely on machine learning algorithms, e.g., voice conversion (VC) [27], voice synthesis (VS) [6, 18], or adversarial example (AE) [15, 56], imposes heavy computational overhead and is inapplicable. Meanwhile, `MicPro` has to strike a balance between anonymity and usability. Prior works that utilize signal processing technologies, e.g., vocal tract length normalization (VTLN) [28, 33, 41, 42], or McAdams Transformation [21, 39], cannot well balance the anonymity and usability, i.e., they produce audios with low anonymity or sometimes sound like robots. To overcome the aforementioned challenges, `MicPro` solves the following two questions.

How to achieve anonymity with existing signal processing techniques? `MicPro` chooses to exploit existing signal processing algorithms because they can be implemented inside most microphone modules. Particularly, we utilize a widely used codec named code excitation linear prediction (CELP) [45] to modify speech signals, which is designed to support low-latency applications and is the basis of many codecs, e.g., G.729 [7], AMR [8], and MPEG-4 [37]. Meanwhile, we discovered that one of the key acoustic features to characterize individual voice biometrics is formants [58]. Formants are determined by the shape of the vocal tract, which differs due to the unique physiological structure of the vocal organs among people. Thus, we propose a voice transformation technique named *formant transformations* by modifying linear spectrum frequencies (LSFs), which are coefficients extracted by CELP and represent the formant distribution of one's speech signal. We investigate three transformation functions that convert original formants to new ones and thereby change the voiceprint features. Since the conversion does not change the speaker's fundamental frequency and harmonic components, it preserves the speech quality to some extent.

How to trade off anonymity and usability? It is difficult to achieve anonymity and usability simultaneously while performing speech anonymization: A stronger anonymity requires a greater level of speech modification, which inevitably results in degradation of usability, e.g., lower speech recognition accuracy and less likely to be human voices. To strike a balance in between, we define usability to be intelligibility and naturalness and formulate multi-objective optimization problems. We solve the problem with Non-dominated Sorting Genetic Algorithm (NSGA-II) and choose feasible solutions from the Pareto Front. We evaluate the `MicPro` performance with three automatic speaker verification (ASV) systems, three automatic speech recognition (ASR) systems, and four corpora. We implement `MicPro` on a Respeaker embedded with G.729 codec to validate its performance. Our experiments show that for the state-of-the-art ASV systems, `MicPro` outperforms existing software-based strategies that utilize signal processing techniques and achieve an EER that is $5 \sim 10\%$ higher and an MMR that is 20% higher while maintaining a similar level of usability.

We summarize our contributions as follows:

- To the best of our knowledge, we propose the first privacy-by-design microphone modules, which can produce anonymous audio recordings with biometric information suppressed while preserving speech quality without hardware modification.
- We design formant transformation algorithms using a generic CELP codec and formulate optimization problems to determine the coefficients that can achieve both anonymity and usability, specifically in virtual calls.
- We implement `MicPro` on an off-the-shelf microphone, validate the effectiveness of `MicPro`, and demonstrate the resistance of `MicPro` to various threats.

2 BACKGROUND

2.1 ASV and ASR

Automatic speaker verification systems (ASVs) and automatic speech recognition systems (ASRs) have been widely used in applications like voice assistants. They can swiftly identify speakers and recognize speech content by utilizing the implicit features hidden in speech signals [26, 29, 34]. These techniques provide a convenient and user-friendly way to control intelligent devices and pass certification with a low cost and acceptable efficiency [2, 51]. For the same speech, ASVs can distinguish the speaker if he or she has been registered, while ASRs can decode the semantic information and convert the acoustic signal into text. This is because any speech spoken by a real speaker conveys both features of the speaker and the content. To achieve better performance, ASVs and ASRs try to separate unrelated information and extract the essential features that are well-suited to their tasks.

For ASVs, classical methods use an acoustic model to extract speaker embeddings which are still widely used today for its high accuracy [14, 47]. In addition, the deep neural network also plays a crucial role in ASVs due to its feasibility of constructing end-to-end systems [16]. Both these two approaches can achieve state-of-the-art performance. For ASRs, classical ASRs consist of several modules like the acoustic model and language model [52]. Recently, end-to-end ASRs have gradually challenged classical ASRs with higher performance and more straightforward implementation [3, 5].

2.2 Threat of Voice Privacy Leakage

With the popularity of ASVs and ASRs, many works have raised concerns about the potential misuse of users' speech audios. One of the most vital threats is that these speech systems may leak users' voiceprint privacy. The nature of speech for communication inevitably forces us to expose our voiceprints to others. However,

these speech systems provide attackers with a convenient way to attain a person's voiceprint information and further compromise privacy security, such as spoofing identity to bypass authentication or recognizing the speaker of sensitive content. To better illustrate the jeopardy of privacy leakage, we define the following two attacks against verification and identification:

- *Identity spoofing attack*: If an attacker knows the victim's identity and has access to audio posted by the victim, she may exploit this audio for a replay attack [55] or a voiceprint mimicry attack [60], which can fool the voiceprint verification of the victim's devices. For example, the attacker and the victim are in the same video conference, and the attacker can record the victim's speech without restriction.
- *Identity inference attack*: If the attacker does not know the victim's identity but has access to many victims' audio without their knowledge, she can easily match the identification of victims by ASVs and further infer their sensitive information. In this situation, we assume the attacker has a pool of potential victims' speech, helping her ASVs for identity inference. For example, the attacker may be an untrusted application or a cloud server that illegally collects the audio of a large number of users [30, 31]. Through identity matching, the attacker can targetedly steal the victim's privacy from big data, such as schedule, health condition, and financial situation.

2.3 Voice Anonymization

To defend against the voice privacy leakage, researchers proposed voice anonymization by hiding one's identity. Some traditional voice anonymization methods change one's voice so that people can not distinguish his or her identity at all, like changing the fundamental frequency. Recent voice anonymization, like speech conversion or speech synthesis, can entirely destroy the original voiceprint and produce a new speech with the same content. However, with the rising of the Internet and social media, people are unwilling to endure the cost of anonymization because they hope their family and friends can distinguish them by their voice. In other words, voice anonymization should hide the user's identity in front of intelligent speech systems instead of human beings. Therefore, an ideal voice anonymization system should not only remove the speaker's information from the original speech to avoid potential attackers stealing privacy, but also meet the requirements of anonymity, intelligibility, naturalness, and acoustic identifiability [15].

Under this anonymization system, ASVs can not distinguish the original speaker. Intelligibility and naturalness confine that both humans and ASRs can understand anonymized speech. Acoustic identifiability seems to contradict anonymity. However, it is required that human beings can distinguish anonymized speech. This constraint widely exists in social media. In conclusion, an ideal speech anonymization should have the ability to deceive speaker identification and verification systems but maintain the original acoustic quality as well as possible.

Existing anonymization works are all implemented by software algorithms and can be divided into three classes, i.e., signal process [39, 53], speech conversion & synthesis [6, 27], and adversarial examples [11, 15]. These methods demonstrate divergent performance

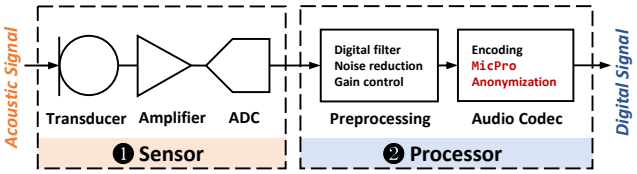


Figure 2: Block diagram of a microphone module. A microphone module consists: (1) a sensor to convert acoustic signal into digital signal; (2) a processor to preprocess and encode the digital signal. MicPro is embedded in the audio codec.

in the above requirements. Above them, anonymization based on adversarial examples can make a good trade-off in all these requirements. However, adversarial examples need a high-consumption computation from the neural network and can only be executed at the software level. It inevitably raises concerns about the transferability and interpretability of adversarial examples, the real-time efficiency of the system, and the security of published software.

2.4 Microphone Module

Smart devices have raised people's awareness because of the emergence of IoT and smart sensing. Specifically, the intelligent voice system is a popular technology because it provides a convenient computer-human interface. The demand for voice interaction makes microphone modules embedded in almost all smart devices. Typically, a microphone module comprises a sensor and a processor, as shown in Fig. 2. The sensor consists of a transducer, amplifier, and analog-digital converter (ADC). The processor is usually a microcontroller or a dedicated digital signal processor (DSP) to process the audio signal. After being recorded by a microphone module, an acoustic signal is converted to a digital signal, and can be stored locally or sent over the network to remote servers.

Audio codec. Compression is necessary for transmitting audio since uncompressed audio consumes a lot of bandwidth. Audio codecs can be divided into three types: *waveform coders*, *vocoders*, and *hybrid coders* [49]. The commonly used coders now are *hybrid coders*, which combine the high quality of waveform coders and the low bit-rate of vocoders. This hybrid coding technique is the so-called code excitation linear prediction (CELP), which merges vector quantization (VQ) with analysis-by-synthesis (AbS) to achieve balanced performance between efficiency and quality [19]. That is why CELP-based codecs have been widely used, especially in voice over Internet protocol (VoIP). Our work, MicPro, is developed on the basis of this codec.

3 THREAT MODEL

3.1 Motivation

Our goal is to anonymize audios in low-latency scenarios such as virtual calls. In these scenarios, previous anonymization schemes encounter two challenges: (1) most anonymization schemes that require an entire piece of recorded audio to be input for anonymization are impractical for the low-latency, frame-by-frame transmissions. Although some SP-based schemes could be adapted for virtual calls, we have shown that they struggle to strike an effective balance between anonymity and usability (in Sec. 6.2), and remain

susceptible to attacks from informed adversaries (in Sec. 6.4.2). (2) The anonymization process becomes futile if an attacker can obtain the original audio before it is anonymized at the software level. For example, an attacker might be able to compromise the victim's device and gain control of the microphone or steal local audio files directly [22, 23]. An attacker could also be an untrusted remote server claiming to provide an anonymization service.

Considering these issues, we envision MicPro working before the software level, i.e., in the microphone module. MicPro can be implemented in low-latency scenarios like virtual calls, with a low overhead that is compatible with microphone hardware with limited processing capabilities. Once produced by MicPro, the audio recordings, regardless of where they are stored, either locally or in the cloud, will protect the voiceprints of users. The additional benefit of this microphone-based approach is that it can be flexibly applied to various software-level applications, such as ASR.

3.2 Defense Goal

We assume that attackers can access the audio from either the Internet or the recorded audio files. Once the audio is obtained, the attackers can extract the voiceprint of the victim and further launch attacks such as *spoofing attack* and *inference attack*. For either attack, MicPro protects the voiceprint privacy from the microphone module level and outputs the privacy-preserving audio to the subsequent applications. MicPro shall meet the following requirements:

- **Anonymity.** Audio output from microphone modules should maintain anonymity to resist privacy leakage threats.
- **Usability.** We should preserve the usability in terms of intelligibility for listeners, low latency for transmissions, and lightweight for implementations.

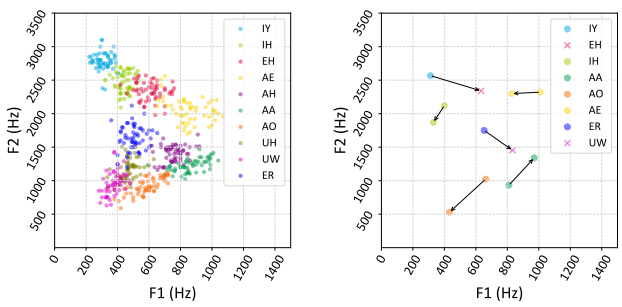
4 PRELIMINARY ANALYSIS

We aim to safeguard voiceprint privacy from the microphone module. Before presenting the design of MicPro, it's essential to assess its feasibility. Here, we showcase that modifying some coefficients of a codec within a microphone module can modify the voiceprint of an audio.

4.1 Change Voiceprint with Microphone Module

The fundamental frequency (F0) and formants (F1, F2, F3, $\cdots$) of audios carry the identity information while changes of formants frequencies convey linguistic information [32]. The fundamental frequency is the basic frequency of vocal cord vibration when voiced (i.e., pronouncing vowels). Formants are related to the shape of the vocal tract, which differs among people due to one's unique physiological structure of the vocal organs. As a result, formants are significant parameters to characterize one's individuality [58]. Both F0 and formants are frequency-domain features that can be displayed in a spectrogram. Therefore, converting a signal sequence to the spectrogram is the first step in calculating various acoustic features, such as Mel-scaleFrequency Cepstral Coefficients (MFCC) which is widely used to train ASR and ASV models.

We use the Peterson-Barney database [12] to illustrate the formant distribution of 10 vowels in English and mark them in Fig. 3(a). The F1 and F2 formants, which are the first and second peak values



(a) Formants of different vowels and different speakers

(b) Formants change after tuning the microphone module parameters

Figure 3: (a) shows the frequency of F1 and F2 formants for vowels (in color) and speakers (in dot). Formants of the same vowel from different speakers tend to cluster within a specific region. (b) shows the effect of adjusting the codec parameters on formant values. This comparison highlights the potential for modifying the voiceprint at the microphone module level.

of the spectrum, are marked for each vowel. The formant frequencies marked with the same color fall within a particular range. The position of the F1 and F2 formants determines the perception of a certain vowel. That's why we can tell that speakers are saying the same word even if they have different timbres.

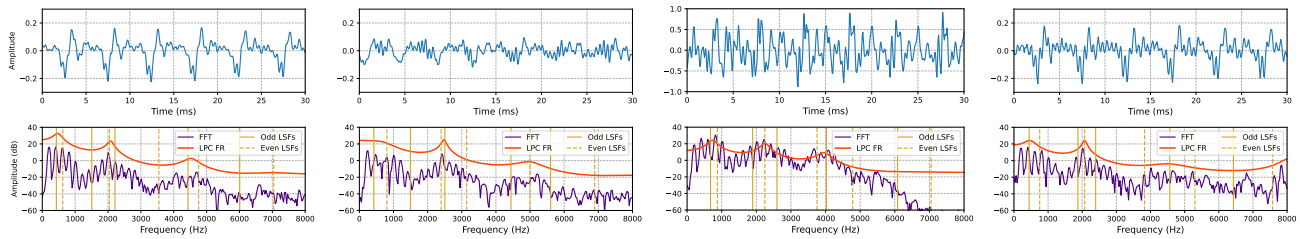
Apart from semantic information, the speaker's individuality is also reflected in the divergence of formant distribution. Therefore, modifying the formants can alter the voiceprint. To investigate this, we randomly modified the linear prediction coding (LPC) coefficients in a CELP codec, which will be explained later, to check for changes in the formants of voiced frames. As shown in Fig. 3(b), the modification of LPC coefficients did change the positions of the formants. However, this random modification resulted in uncontrollable changes in formant positions, leading to speech recognition or verification errors (marked with crosses). In some cases, the shifted formants caused vowel perception errors, leading to a loss of intelligibility. Nevertheless, this preliminary validation demonstrates the feasibility of modifying formants by tuning the microphone module parameters. In the following sections, we will elaborate on how to achieve controllable privacy protection by adjusting these parameters.

Remark: It is feasible to modify the voiceprint of an audio from the microphone module level by changing the LPC coefficients.

4.2 Change Voiceprint by Formant Modification

In this subsection, we will explain how to modify the voiceprint by manipulating formant features. We first provide the basis of LPC analysis, and then describe the methods for changing formants using line spectrum frequencies (LSFs) based on LPC.

LPC analysis. LPC has emerged as a powerful technique for speech signal processing, including speech feature extraction and speech synthesis [1]. The fundamental principle of LPC is based on the approximation of an autocorrelation sequence at any time instant, which can be represented as a linear combination of its past p sample values and a residual $e(n)$, as shown in the following



(a) Original frame (b) VoiceMask frame (c) McAdam Transformation frame (d) V-Cloak frame

Figure 4: The waveform and spectrum of a voiced speech *a*. FFT: Fast Fourier Transform. LPC FR: LPC filter frequency response, also the envelope of FFT. Odd LSFs: Odd order of LSFs ω_i. Even LSFs: Even order of LSFs θ_i. LPC FR shows the envelope of the FFT spectrum, and formants are shown as the peak values sandwiched by a pair of narrow lines.

equation:

$$\hat{x}(n) = -\sum_{k=1}^{p} a_k x(n-k) + e(n) \tag{1}$$

where a_k is called LPC coefficient, and p is the LPC order. By applying Z-transform to Eq. 1, we have:

$$X(z) = E(z)/A(z) \tag{2}$$

where $1/A(z) = 1/(\sum_{k=0}^{p} a_k z^{-k})$ is the LPC filter. LPC is adaptable for short-time speech analysis because the speech signal has components of different frequencies and is considered stationary over a short period of time (10ms to 30ms). The spectrum of $1/A(e^{j\omega})$ represents the envelope of the short-time Fourier transform (STFT) of $x(n)$. The peak values of the envelope indicates the formant positions. LPC analysis decomposes the speech signal into a residual signal $E(z)$ (for glottal excitation when voiced) and a channel filter $1/A(z)$ (for the vocal tract response). LPC analysis can effectively encode the time-domain and frequency-domain features of the original speech signal with a few coefficients [20]. In addition, LPC can also provide an accurate speaker vocal tract model, which is significant for speaker recognition and speech synthesis.

Line spectrum frequencies. A disadvantage of the LPC coefficients is that small changes in the coefficients may make the LPC filter unstable [43], i.e., the filter appears to have poles outside the unit circle of the complex plane. Therefore, it is undesirable to quantize LPC coefficients directly in speech coding but to replace them with equivalent coefficients, called line spectrum pairs (LSPs) or line spectrum frequencies (LSFs). LSPs are derived from the denominator polynomial $A(z)$ of the LPC filter. They are calculated as follows [35].

Let $P(z) = A(z) - z^{-(p+1)}A(z^{-1})$ and $Q(z) = A(z) + z^{-(p+1)}A(z^{-1})$. The zeros of these two polynomials are all on the unit circle [48], thus they can be written by:

$$\begin{aligned} P(z) &= (1+z^{-1})\prod_{i=1}^{p/2}(1-2\cos\omega_i z^{-1}) \\ Q(z) &= (1-z^{-1})\prod_{i=1}^{p/2}(1-2\cos\theta_i z^{-1}) \end{aligned} \tag{3}$$

where p is the LPC order, ω_i, θ_i are the so-called LSFs, and $\cos\omega_i$, $\cos\theta_i$ are LSPs. The zeros of $P(z)$ and $Q(z)$ are interlaced with each other, that is:

$$0 < \omega_1 < \theta_1 < \cdots < \omega_{p/2} < \theta_{p/2} < \pi \tag{4}$$

LSFs are considered to reflect the characteristics of the LPC spectrum. Precisely, a pair of ω_i and θ_i determine the position and amplitude of a formant. The closer they are, the higher the amplitude of the formants here. In addition, LSFs facilitate interpolation in speech encoding. As long as the sequence order of LSFs is not changed, the synthesized LPC filter can be guaranteed to be stable [35], which can be easily obtained by $A(z) = 1/2[P(z) + Q(z)]$. In the following, we utilize this property to achieve controllable transformation of the formants.

Remark: LSFs can be used to model formants as a stable and alternative representation for LPC coefficients. We can modify LSFs to reshape the formant features for voiceprint modification.

4.3 Modification of Formants in Previous Works

In this subsection, we perform LPC analysis for previous anonymization works to further corroborate the formant changes in anonymized speech. We investigate two SP-based works, McAdam Transformation (MT) [39] and VoiceMask (VM) [42], as well as an ML-based work, V-Cloak [15]. To analyze the modification in signal features of these anonymization methods, we take the vowel frames *a* from *librispeech-test-clean* [38] as an example of a voiced speech, and visualize their time-domain waveforms and frequency-domain responses, as shown in Fig. 4. Figure 4(a) shows the original waveform, FFT, LPC filter frequency response, and LSFs. The LPC order is set to $p = 10$. The F1 and F2 formants are located in the frequency band of $[0, 4000]$Hz, with frequencies of about 500 and 2000 Hz.

VoiceMask (VM). The VM modifies the frequency components of the original audio signal by a bilinear function:

$$f(\omega, \alpha) = |-j\ln\frac{e^{j\omega}-\alpha}{1-\alpha e^{j\omega}}| \tag{5}$$

where $\omega \in [0, \pi]$ is the digital frequency (calculated by FFT), and $\alpha \in (-1, 1)$ is the warping factor that indicates the strength of frequency warping. Fig. 4(b) shows a VM frame of the vowel. The conversion destroys the harmonic property of the frequency components. Therefore it loses its naturalness compared to a human voice. We can also see a slight shift in the formants. That's because the formants follow the frequency to distort to a new position.

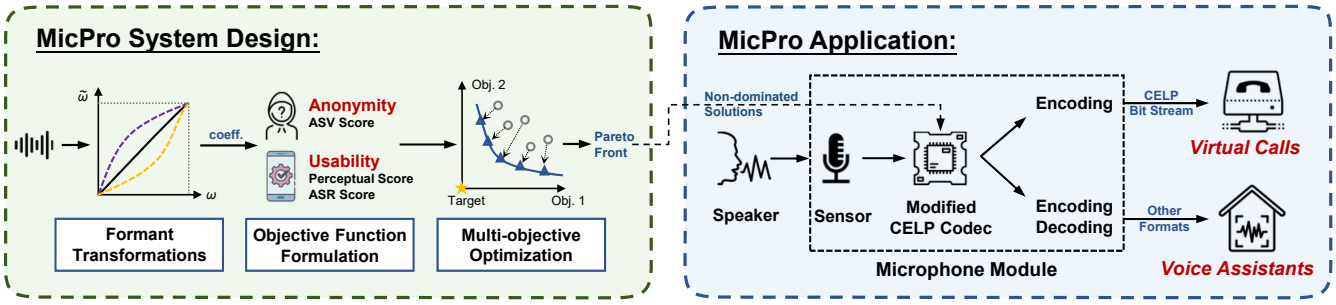


Figure 5: MicPro Overview. We define the formant transformation functions and formulate a multi-objective function for specific privacy-preserving tasks. To solve the problem, we employ an improved genetic algorithm to get a set of non-dominated solutions. The available solutions are embedded into a modified CELP codec inside a microphone module to enable privacy-preserving applications such as virtual calls or voice assistants.

McAdam Transformation (MT). The MT modifies the pole angle of the LPC filter, where the poles are the conjugate complex roots of $A(z)$:

$$A(z) = \prod_{i=1}^{p/2}(1 - r_i e^{\mathrm{j}\theta_i} z^{-1})(1 - r_i e^{-\mathrm{j}\theta_i} z^{-1}) \tag{6}$$

where r_i is the pole radius, θ_i is the pole angle, and j is the imaginary unit. MT uses a coefficient $\alpha \in [0.5, 1]$ to modify the pole angle by $\theta_i' = \theta_i^{\alpha}$, when $\alpha = 1$ means no modification and $\alpha < 1$ causes a shift in the resonant frequency of LPC filters. The default value of α is typically set to 0.8. Fig. 4(c) shows an MT frame of the vowel. This transformation tends to centralize the formants and concentrate the energy towards the intermediate frequency. It can also be reflected in the time domain waveform, where the amplitude of high-frequency oscillation increases significantly. Note that MT only changes the energy distribution of frequency components, while the fundamental frequency and its harmonics shown in the FFT spectrum remain unchanged.

V-Cloak. V-Cloak is an anonymization method based on adversarial learning. It adds delicate and imperceptible perturbations to the original signal, which can be regarded as an adversarial example. This method makes fine-grained modifications to the time-domain waveform and can be observed from Fig. 4(d). Features in the frequency domain have no noticeable changes, while perturbations are reflected by an increase in high-frequency energy.

To summarize, SP-based anonymization methods essentially modify the frequency features of the signal. MT changes the spectrum envelope while VM changes the frequency components. Both methods inevitably lead to a certain degradation in usability. The two SP-based methods utilize constrained coefficients for the transformation, which limits their ability to make fine-grained modifications to the original signal. Moreover, a radical change in the signal can result in intelligibility degradation. Therefore, we aim to address two critical issues in the design of MicPro. Firstly, we investigate whether there exist SP-based modifications, such as formant transformation, that can preserve the usability of the audio as much as possible. Secondly, we aim to determine the extent to which such modifications can be made without negatively impacting the usability.

5 DESIGN

In order to achieve voiceprint privacy protection, MicPro leverages the built-in functions of the microphone module to modify formant features. MicPro first transforms the LSFs of the original audio signal to achieve anonymity. The transformation functions are then applied to the CELP encoder to fulfill the audio transformation with only minor modifications.

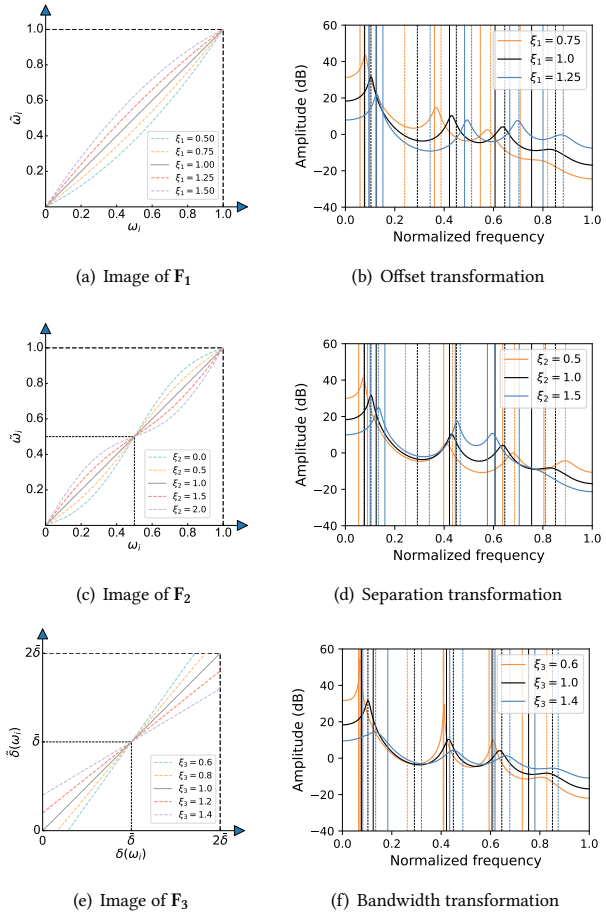
5.1 Overview of MicPro

We present the overview of MicPro in Fig. 5. First, we define the formant transformations to modify the formant features of the original audio frames, as described in Sec. 5.2. These transformations are parameterized using three coefficients that govern the form of the transformation. Due to the difficulty of obtaining optimal coefficients, we formulate a multi-objective optimization problem in terms of different defense tasks, as outlined in Sec. 5.3. We use the ASV score to indicate anonymity, and perceptual score and ASR score to indicate usability. To solve this problem, we adopt a multi-objective optimization algorithm based on Genetic Algorithm (GA), which enables us to obtain the Pareto front, as described in Sec. 5.4.

Figure 5 also depicts the process of protecting voiceprint privacy using a modified CELP codec with a microphone module. The codec applies formant transformations to the original audio frames with pre-trained coefficients. In applications such as virtual calls, the microphone module encodes the original audio signal into a bit stream for transmission over the Internet. In other applications such as voice assistants, the microphone performs the entire encoding and decoding process to anonymize voice commands.

5.2 Formant Transformations

In Sec. 4, our findings demonstrate a strong correlation between the distribution of formants and speech content, as exemplified by the vowel sounds shown in Fig. 3(a). Furthermore, our analysis reveals that, for a given vowel sound, significant variability in the F1 and F2 formants exists among different speakers, implying that formant characteristics can be used to distinguish between speech content and speaker identity. Given this insight, it follows that the manipulation of formant distribution can serve as a means of



(a) Image of $\mathbf{F_1}$ (b) Offset transformation

(c) Image of $\mathbf{F_2}$ (d) Separation transformation

(e) Image of $\mathbf{F_3}$ (f) Bandwidth transformation

Figure 6: Images of the three formant transformation functions and the respective spectrum changes. The right-side sub-figures demonstrate examples of the audio spectrum after the transformation.

altering speaker characteristics while minimizing the distortion of textual information.

5.2.1 Define the transformation functions. In Sec. 4.2, we discuss the ability to reshape formants through the manipulation of LSFs. We hereby define transformation functions for LSFs, which can be used to produce a desired pattern of formant changes. Previous works have explored the use of LSF transformations for speech enhancement [35] and voice conversion [58]. The former utilizes LSF operations to modify formant bandwidths and positions, while the latter applies a piecewise linear function to transform the LSFs of a source speaker to those of a target speaker. In line with this, we design the following transformations, specifically, F_1 and F_2 to relocate the positions, and F_3 to adjust the bandwidths.

F_1: Formant offset function. Firstly we consider the overall offset of the formants. As formants are spectrum peaks, an offset of formants shifts spectral energy toward higher or lower frequencies. This function is formulated as follows:

$$\tilde{\omega_i} = F_1(\omega_i, \xi_1) = \omega_i + \omega_i(\xi_1 - 1)(1 - \omega_i) \quad i = 1, \cdots, p \tag{7}$$

where $\omega_i \in [0, 1]$ is the ith normalized LSF, p is the order of LPC analysis. We refer ξ_1 as *offset coefficient*. The image of F_1 and a schematic diagram of spectrum change are shown in Fig. 6(a) and Fig. 6(b). An offset coefficient of $\xi_1 > 1$ shifts the formants towards higher frequencies, while $\xi_1 < 1$ shifts them towards lower frequencies.

F_2: Formant separation function. The second is the separation function which modifies the degree of dispersion between the formants. It can be expressed as follows:

$$\tilde{\omega_i} = F_2(\omega_i, \xi_2) = \omega_i + (\xi_2 - 1)\sin(2\pi\omega_i)/p \quad i = 1, \cdots, p \tag{8}$$

We refer ξ_2 as *separation coefficient*. The image of F_2 and a schematic diagram of spectrum change are shown in Fig. 6(c) and Fig. 6(d). $\xi_2 > 1$ means to gather the formants, while $\xi_2 < 1$ means to spread the formants.

F_3: Bandwidth adjustment function. Finally, we consider modifying the bandwidth of the formants. This can be achieved by adjusting the separation of adjacent lines. We additionally define $\omega_0 = 0$ and $\omega_{p+1} = 1$. The distance of the two adjacent lines is:

$$\delta(\omega_i) = \omega_{i+1} - \omega_i \quad i = 0, \cdots, p \tag{9}$$

Adjust the distance by:

$$\tilde{\delta}(\omega_i) = \delta(\omega_i) + (\xi_3 - 1)[\bar{\delta}(\omega_i) - \delta(\omega_i)] \tag{10}$$

where $\bar{\delta}(\omega_i) = \sum_{i=0}^{p} \delta(\omega_i)/(p+1) = 1/(p+1)$ is the average lines distance. We refer ξ_3 as *expansion coefficient*. $\xi_3 > 1$ means expanding the formants bandwidth while $\xi_3 < 1$ means shrinking the formants bandwidth. The modified LSFs are given by:

$$\tilde{\omega_i} = \sum_{k=0}^{i-1} \tilde{\delta}(\omega_k) \quad i = 1, \cdots, p \tag{11}$$

Then F_3 can be written by:

$$\tilde{\omega_i} = F_3(\omega_i, \xi_3) = \sum_{k=0}^{i-1} \{\omega_{k+1} - \omega_k + (\xi_3 - 1)[\frac{1}{p+1} - \omega_{k+1} + \omega_k]\} \tag{12}$$

The image of F_3 and a schematic diagram of spectrum change are shown in Fig. 6(e) and Fig. 6(f). It can be observed that expanding the formant bandwidth is equivalent to flattening the spectrum.

These three transformation functions are cascaded to combine a complex transformation, which is given by:

$$\mathcal{F}(\tilde{\omega_i}, \xi) = F_3\big(F_2\big(F_1(x, \xi_1), \xi_2\big), \xi_3\big) \tag{13}$$

5.2.2 Stability and invertibility analysis. To ensure the stability of the LPC filter after transforming, we cannot change the ordering of the LSFs. In other words, we should guarantee that $\tilde{\omega_1} < \tilde{\omega_2} < \cdots < \tilde{\omega_p}$. A sufficient condition for it is to make the functions monotonically increasing. To do so, we constrain the coefficients $\xi_1, \xi_2, \xi_3 \in [0, 2]$. As a result, $\forall \omega_i \in [0, 1]$ we have:

$$\frac{\partial F_1(\omega_i, \xi_1)}{\partial \omega_i} = 2(1 - \xi_1)\omega_i + \xi_1 \geq 0 \tag{14}$$

$$\frac{\partial F_2(\omega_i, \xi_1)}{\partial \omega_i} = 1 + \frac{2\pi(\xi_2 - 1)}{p}\cos(2\pi\omega_i) \geq 1 - \frac{2\pi}{p} > 0 \tag{15}$$

$$\frac{\partial F_3(\omega_i, \xi_1)}{\partial \omega_i} = 2 - \xi_3 \geq 0 \tag{16}$$

Thus, the transformation functions are monotonically increasing, and our transformations will not induce instability of the filter.

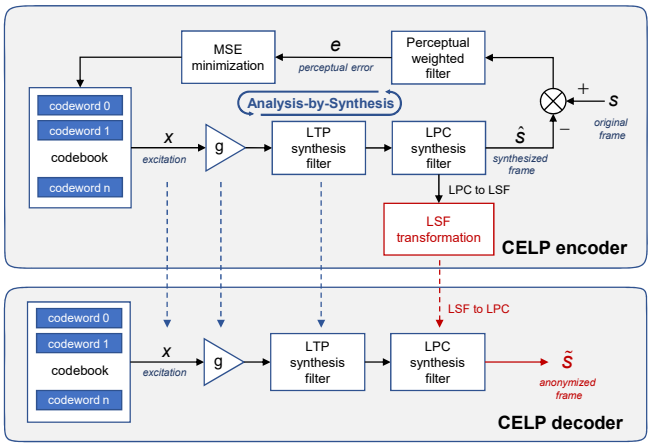


Figure 7: The LSF transformations are embedded in a generic code excitation linear prediction (CELP) codec. CELP encoder first performs Analysis-by-Synthesis (AbS) for each audio frame. The codebook excitation x, excitation gain g, LTP coefficients, and LPC coefficients are obtained by AbS. LPC coefficients are converted to LSFs for quantization. The LSF transformation functions are only applied to the LPC synthesis filter to achieve the transformation.

We point out that these functions are invertible, an essential property to the security of our anonymization method, which is depicted in Sec. 6.4. Since monotonicity has been proved, and the definition and value domains are known as $\omega_i \in [0, 2]$, $\tilde{\omega}_i \in [0, 2]$, we can say they are bijection, i.e., invertible. Let $F^{-1} : \tilde{\omega}_i \rightarrow \omega_i$ denotes the inverse functions of transformation functions.

For F_1 we have its inverse function as:

$$\omega_i = \frac{\xi_1 - \sqrt{\xi_1^2 - 4(\xi_1 - 1)\tilde{\omega}_i}}{2(\xi_1 - 1)} \tag{17}$$

For F_2, its inverse function $\omega_i = F_2^{-1}(\tilde{\omega}_i, \xi_2)$ has no explicit functional expression, as it is a transcendental equation. Instead, we can obtain the inverse function value by a look-up table.

For F_3, we first calculate $\delta(\omega_i)$ using Eq. 10:

$$\delta(\omega_i) = \frac{\delta(\tilde{\omega}_i) - (\xi_3 - 1)\bar{\delta}(\omega)}{2 - \xi_3} \tag{18}$$

Then the inverse function is given by:

$$\omega_i = \sum_{k=0}^{i-1} \delta(\omega_i) = \sum_{k=0}^{i-1} \frac{\tilde{\omega}_{i+1} - \tilde{\omega}_i - (\xi_3 - 1)/(p+1)}{2 - \xi_3} \tag{19}$$

5.2.3 Modify the CELP codec for formant transformations. The premise of being able to apply the formant transformations is that we have already obtained the LSFs. However, extracting LSFs is complex, involving several steps such as speech framing, windowing, and LPC analysis. Additionally, the accuracy of LSFs can be affected by the choice of analysis parameters used in these steps. To simplify the process, we utilize the CELP codec to decrease the computational overhead of these steps.

CELP is a speech compression technique that combines LPC analysis and waveform coding, as shown in Fig. 7. In a generic CELP codec, LPC analysis and long-term analysis are performed to model the formants and pitch, respectively. Then the analysis-by-synthesis (AbS) process is executed to determine an optimal excitation from a fixed codebook [43]. LSFs are used in the CELP codec as an equivalent mathematical representation of LPC coefficients because: (1) checking the stability and fine-tuning LSFs can easily eliminate the instability of the LPC filter caused by quantization errors [24]. (2) They may be quantized for fewer bits compared with other coefficients while maintaining the speech quality [35].

We leverage these inherent properties of LSFs in the CELP codec to reduce the overhead of signal preprocessing in `MicPro` and improve the quality of the synthesized speech. In the CELP encoder, the codebook index, excitation gain, LTP filter coefficients, and LPC filter coefficients are quantized to bitstream for transmission. In the CELP decoder, the bitstream file is unpacked to the above coefficients used for frame reconstruction. Our modification is applied before the LSF quantization step and is highlighted in red in Fig. 7. The modified LSFs are also restored to new LPC coefficients, which are used for AbS later.

5.3 Objective Function Formulation

5.3.1 Formulate the privacy-preserving problem. Here we show how to determine the coefficients ξ to achieve anonymization by formulating the problem as an optimization problem. Specifically, given an original audio signal $x = [x_1, \cdots, x_N] \in \mathbb{R}^{1\times N}$, we aim to find a transformed audio signal $\tilde{x}$ which destroys the original voiceprint within x. Typically, a voiceprint can be extracted by ASVs that output a speaker's identity represented by an embedding vector $v(x) \in \mathbb{R}^{1\times V}$. We define the objective function to be optimized as follows:

$$\begin{aligned} \min_{\xi} \quad & S_{\text{ASV}}[v(x), v(\tilde{x})] \\ \text{s.t.} \quad & x, \tilde{x} \in [-1, 1] \quad \text{and} \quad \xi \in [0, 2] \end{aligned} \tag{20}$$

where $S_{ASV}[\cdot, \cdot]$ denotes the similarity between two voiceprint vectors calculated by an ASV system, and $\tilde{x}$ is the transformed audio whose LSFs are $\tilde{\omega}_i = \mathcal{F}(\omega_i, \xi)$. We assume that both the original signal x and transformed signal $\tilde{x}$ are limited to $[-1, 1]$ in accordance with the standard of PCM format.

5.3.2 Augment the objective function. After formulating our task as an optimization problem, we need to refine the objective function to consider audio usability. Specifically, the output audio signal should maintain its intelligibility and naturalness. In addition, to balance the trade-off between anonymity and usability, traditional approaches tend to use the weighted sum of the usability and anonymity scores. However, it is not easy to choose appropriate weights since anonymity and usability are different metrics.

To address these issues, we augment the objective function by introducing intelligibility and naturalness assessment. Based on our defense tasks defined in Sec. 6.2, we use perception score in **T1** (for human listeners in virtual calls) and ASR score in **T2** (for ASRs). Moreover, instead of using a weighted sum, we formulate multiple objective functions to avoid the need for weight selection as follows:

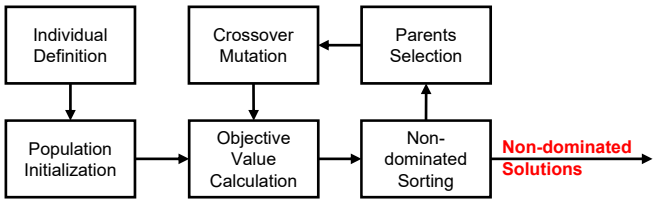


Figure 8: Flow chart of MicPro NSGA-II.

$$\begin{aligned} \textbf{T1}: \min_{\xi} \quad & S_{\text{ASV}}[v(x), v(\tilde{x})], S_{\text{pept}}(x, \tilde{x}) \\ \textbf{T2}: \min_{\xi} \quad & S_{\text{ASV}}[v(x), v(\tilde{x})], S_{\text{ASR}}(x, \tilde{x}) \\ \text{s.t.} \quad & x, \tilde{x} \in [-1, 1] \quad \text{and} \quad \xi \in [0, 2] \end{aligned} \tag{21}$$

where $S_{\text{pept}}(\cdot,\cdot) = 1 - \text{STOI}$ is the perception quality score indicating the audibility for human beings and $S_{\text{ASR}}(\cdot,\cdot)$ is the ASR score measured in terms of word error rate (WER). STOI is short-time objective intelligibility [50].

5.4 Multi-objective Optimization

We consider utilizing an optimization algorithm to solve the problem mentioned above. Considering that the vector quantization in the CELP codec is unsuitable for gradient back-propagation, we regard the problem as a black-box multi-objective optimization problem. We tried several commonly-used algorithms and compared their performance, which can be found at our homepage[1], and finally determined to employ the Non-dominated Sorting Genetic Algorithm (NSGA-II) [13], a multi-objective genetic algorithm based on Pareto optimality. The flow chart of NSGA-II is shown in Fig. 8 and we highlight the main steps in the following:

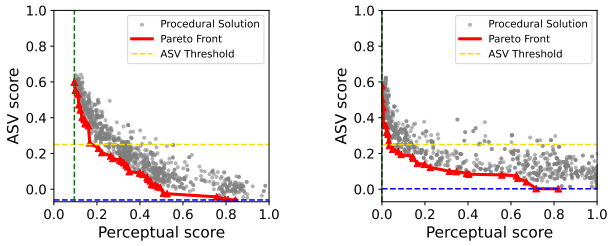
- **Individual definition.** We regard the coefficients as the genes of each individual:

$$g^{(k)} = (\xi_1^{(k)}, \xi_2^{(k)}, \xi_3^{(k)}) \quad k = 1, \cdots, P \tag{22}$$

  where $g^{(k)}$ denotes the genes of kth individual and P is the population size. We set $P = 100$ in our problem.
- **Population initialization.** We randomly initialize the population in the coefficients definition space $[0, 2]^{1\times 3}$.
- **Objective value calculation.** To obtain accurate scores for the objective function, we need to calculate S_{ASV}, S_{pept} and S_{ASR} for each individual using multiple speech samples. However, iterating through the entire training set for each individual can be time-consuming. To alleviate this burden, we randomly select a subset of samples from the training set for each individual. Therefore, we calculate the mean over the selected subset as the expected scores:

$$\mathbb{E}[S^{(k)}] = \frac{1}{N}\sum_{i=1}^{N} S^{(k)}(x_i, \tilde{x}_i) \quad x_i \in \mathcal{D}_{sub} \subset \mathcal{D} \tag{23}$$

  where $\mathcal{D}$ is the training set and $\mathcal{D}_{sub}$ is the subset of $\mathcal{D}$. Without loss of generality, we randomly select 20 pieces of audio for $\mathcal{D}_{sub}$ for each individual in each epoch.
- **Non-dominated sorting.** In this step, NSGA-II algorithm performs fast non-dominated sorting, calculates crowding degree, and applies elite strategy for each individual to generate new parents.
- **Parents selection, crossover and mutation**. This step is the same as basic genetic algorithms. New offspring and parents are combined as the new population for the next epoch.



(a) Optimization results of ASV and perceptual score

(b) Optimization results of ASV and ASR score

Figure 9: Optimization results of NSGA-II. The green and blue dashed lines are the ideal minimums of objective values. The yellow dashed line is the threshold of the ASV score that we adopt. The red curve shows the Pareto Front, i.e., the non-dominated solutions.

We use an established optimization toolkit [9] to implement the above algorithm. We visualize the optimization result in Fig. 9. The red curves are the final results of NSGA-II, i.e., Pareto Fronts, where each red triangle marks the objective values of a set of non-dominated solutions. The grey dots correspond to the procedural results. The green and blue dashed lines indicate the ideal minimum values of objective values. The yellow dashed line is the threshold of the ASVs (0.25 for ECAPA-TDNN). An ASV score below this threshold indicates successful anonymization. To preserve usability, we set the perceptual score threshold to 0.25 and the ASR score threshold to 0.1. The feasible solutions are those that fall below both the ASV and perceptual (or ASR) score thresholds. Finally, we use the coefficients corresponding to a feasible solution to evaluate the performance of `MicPro`.

5.5 Implementation

5.5.1 Software Implementation. To implement `MicPro`, we developed a prototype of the CELP codec in the G.729 codec [7]. G.729 is based on CS-ACELP (Conjugate-Structure Algebraic-Code-Excited Linear Prediction) and possesses all properties of a generic CELP codec. We obtain G.729 source code from the official website of ITU-T [25]. To apply LSF transformation, we investigate the code structure and add a new source code `lsf_trans.c` where we define the transformation functions `Lsf_trans()`. We insert `Lsf_trans()` between `Lsp_Lsf2()` and `Lsp_qua_cs()`. `Lsp_Lsf2()` is the function to convert LSPs to LSFs, and `Lsp_qua_cs()` is the function to quantize LSFs. Both of them are defined in `qua_lsp.c`.

5.5.2 Hardware Implementation. The CELP codec itself is designed for real-time voice communication, and it processes voice signals frame by frame. CELP codec based on C language can be well-compatible with embedded devices, such as DSPs and microcontrollers. We present an example of deploying on a Respeaker Core V2. The microphone module hardware setup can be referred to Fig. 10. More details can be found at our homepage[1].

Dataset	Subset	#Speaker	#Utterance	Duration (s)
VoxCeleb1 (**E**)	*dev*	1,211	148,642	3.9 ~ 144.9
LibriSpeech (**E**)	*train-clean-360*	921	104,014	1.1 ~ 29.7
VoxCeleb1 (**E**)	*test*	40	4,874	3.9 ~ 69.1
LibriSpeech (**E**)	*test-clean*	40	2,260	1.3 ~ 35
VCTK (**E**)	*wav48*	40*	2,000†	2.1 ~ 15.1
AISHELL (**C**)	*test*	20	1,000†	1.9 ~ 14.7

Table 1: Datasets for training and evaluation. *: we use the first 40 speakers in VCTK. †: we randomly select 50 utterances from each speaker. E: English; C: Chinese.

ASV Model	Catagory	EER	ASR Model	Language	WER
ECAPA-TDNN	DNN-based	0.7%	transformer	**E&C**	2.27%
X-Vector	DNN-based	2.5%	wav2vec2	**E**	1.90%
I-Vector	Statistic	2.8%	crdnn-rnn	**E**	3.90%

Table 2: ASVs and ASRs for evaluation. E: English; C: Chinese.

6 EVALUATION

In this section, we conducted a series of experiments to evaluate the performance of MicPro in two fields, i.e., usability and anonymity. Our evaluation compared the performance of MicPro and other baselines according to two different tasks defined in Sec. 6.2. Our proposed method can provide various solutions according to the Pareto front shown in Fig. 9 to fit different tasks better. We choose two existing voice anonymization systems, Mcadam [39] and Voicemask [42], and make a comprehensive comparison with our system in four datasets. In addition, we explore some implicit impacts on performance to verify the robustness of MicPro. A physical domain experiment is also conducted to ensure the feasibility of implementing microphone hardware.

6.1 Experienment Setup

6.1.1 Datasets. We evaluated the performance and transferability of MicPro using four different datasets. Our training dataset is VoxCeleb1 [36] and LibriSpeech [38] (subset *dev* and *train-clean-360*), two English speaker identification datasets. In addition, we use four other widely-used datasets: VoxCeleb1 (subset *test*), LibriSpeech (subset *test-clean*), VCTK (subset *wav48*) [57], and AISHELL (subset *test*) [10] for evaluation. We have provided the details of the datasets used in Tab. 1 We also use our hardware implementation to record direct and indirect human speech (i.e., via a loudspeaker) with 22 participants and 336 utterances each.

6.1.2 Baselines. In comparing MicPro with the baselines, we examine two commonly-used anonymization methods based on signal processing, i.e., McAdam [39] and VoiceMask [42]. To process the test set audios, we apply a fixed McAdam coefficient of 0.8 and a fixed VoiceMask warping factor of 0.1, following the implementations provided in the respective Github repositories [54, 59].

6.1.3 ASVs and ASRs. To evaluate the performance of MicPro, we use three widely-used ASV systems and three ASR systems, which are listed in Tab. 2. In addition to DNN-based ASVs like ECAPA-TDNN [16], we also select traditional statistical-model-based ASVs like I-Vector [14]. As for ASRs, we use three different ones for English datasets and one for Chinese datasets. We obtain the pretrained ECAPA-TDNN model and all ASR models from SpeechBrain [44], while X-Vector and I-Vector models are provided by Kaldi [40].

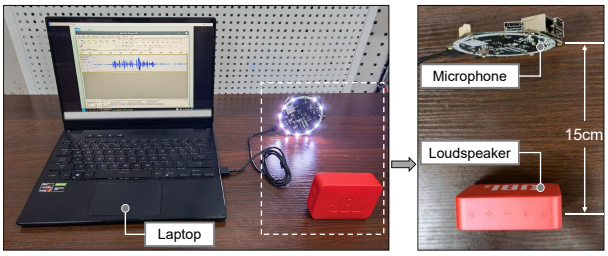
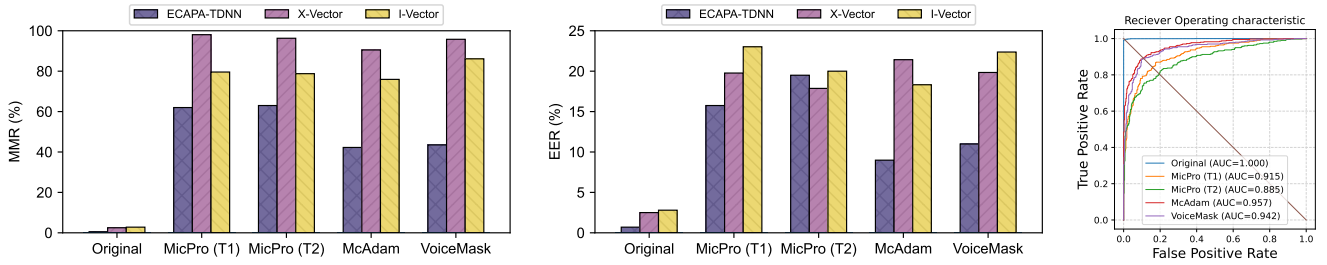


Figure 10: Pysical setup. We use a bluetooth loudspeaker to represent a real speaker. We deploy MicPro on the microphone module to anonymize the recorded audio locally in real time.

6.1.4 Evaluation Metrics. In our evaluation, we employ widely-accepted metrics in the field of anonymization to assess the effectiveness of MicPro. For anonymity, we primarily focus on Mis-Match Rate (MMR) and Equal Error Rate (EER). MMR is more appropriate to the goal of anonymization against speaker verification, while EER provides a comprehensive measure of the anonymization performance against speaker identification. For usability, we adopt Short-Time Objective Intelligibility (STOI), Latency, as well as subjective quality as metrics. These metrics are mainly used to indicate the usability of virtual applications. To illustrate a wider usage range of MicPro, we also choose Word Error Rate (WER) to indicate the usability in ASRs.

- **Miss-Match Rate (MMR).** MMR indicates the probability that anonymized audio cannot be matched with the correct speaker by an ASV system.
- **Equal Error Rate (EER).** EER indicates the rate of an ASV system at which False Accept Rate (FAR) equals False Rejection Rate (FRR).
- **Latency.** Latency indicates the real-time nature of a speech codec and has a great impact on call quality. It is measured in milliseconds (ms).
- **Short-Time Objective Intelligibility (STOI).** STOI [50] indicates speech intelligibility. Its range is quantified from 0 to 1 to represent the percentage of words that are correctly understood.
- **Subjective quality**. We conducted a subjective evaluation of the quality of anonymized audios through a user study. Specifically, we mixed different anonymized audios and recruited 32 participants to rate them on a scale from 0 (worst) to 5 (best). We asked participants to pay attention to four indicators: (1) the intelligibility of the audio, (2) the naturalness of the audio, (3) the similarity of the audio to the original, and (4) whether they would accept such a voice change if they had anonymity needs.
- **Word Error Rate (WER).** WER is used to quantify the dissimilarity between the ASR results obtained from the original audio and the anonymized audio, which is given by:

$$WER = \frac{N_{sub} + N_{del} + N_{ins}}{N_{sum}}, \tag{24}$$



(a) MMR (b) EER (c) ROC curves

Figure 11: The anonymity performance of anonymization systems. (a) & (b): MMR and EER of MicPro and baselines in three ASV models. (c): ROC curves of MicPro and baselines in ECAPA-TDNN.

t_{dur} (s)	t_{enc} (ms)	$\tilde{t}_{enc}$ (ms)	l (ms)	$\tilde{l}$ (ms)	Δl (ms)	$\delta l(\%)$
5	683 ± 18	685 ± 10	16.366	16.370	0.004	0.02
30	3,864 ± 22	3,868 ± 24	16.288	16.289	0.001	0.01
60	7,667 ± 24	7,663 ± 14	16.278	16.277	−0.001	−0.01
120	15,289 ± 45	15,293 ± 32	16.274	16.274	0.000	0.00
Avg.	-	-	16.302	16.303	0.001	**0.01**

Table 3: Latency increase of MicPro. t_{dur}: duration. t_{enc}: encoding time of original codec. $\tilde{t}_{enc}$: encoding time of modified codec. l: latency of original codec. $\tilde{l}$: latency of modified codec. Δl: latency difference. δl: relative latency difference. The results show that the modified codec does not introduce significant additional latency.

where N_{sub}, N_{del}, and N_{ins} are the number of substituted words, deleted words, and inserted words, respectively. N_{sum} is the real word number of the ground-truth audio.

6.1.5 Physical Setup. To validate the feasibility of hardware deployment, we implement MicPro on Respeaker Core V2, a microphone development platform with six microphone arrays and an RK3229 microprocessor running a Linux operating system [46]. The physical setup is shown in Fig. 10. The laptop powers the microphone and controls its recording. A Bluetooth loudspeaker is placed 15cm away from the microphone to represent a human speaker. We embed the modified G.729 codec into the microphone module so that it can record and anonymize audio locally in real time.

6.2 Overall performance

In this part, we evaluate our systems in two different tasks. **Task 1 (T1):** In virtual applications such as calls and meetings, MicPro should conceal the voiceprint of users but keep the naturalness and intelligibility of the speech. **Task 2 (T2):** MicPro should protect users from voiceprints leakage when interacting with ASR systems. In both tasks, MMR and EER are the goals of anonymity optimization. As for Usability, we use STOI and Subjective quality to optimize **Task 1** and use WER to optimize **Task 2**. Note that the goal of MicPro is mainly on **Task 1**.

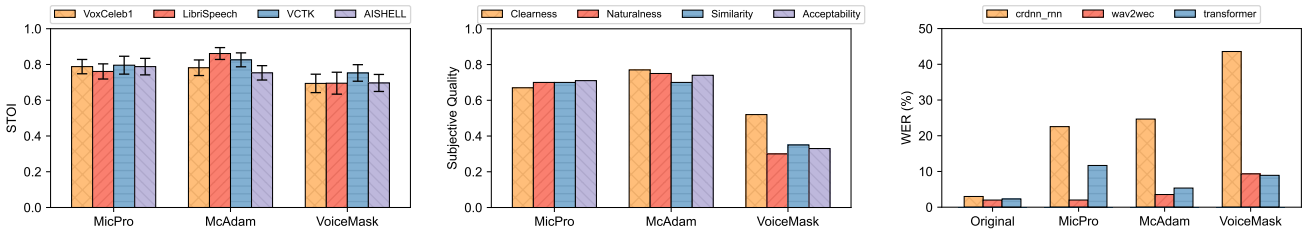
6.2.1 Anonymity. Fig. 11(a) and 11(b) show the result of the comparison between MicPro and baselines. MicPro (T1) and (T2) are two sets of parameters with different optimization goals in usability. We can find that MicPro (T1) can reach a high MMR and EER in all three ASVs. However, the traditional SP base anonymization systems, i.e., Mcadam and Voicemask, can not achieve acceptable performance in the state-of-the-art DNN-based ASVs, ECAPA-TDNN. At the same time, MicPro can still keep an MMR of 63% and an EER of 15%. Even in traditional statistical model base ASVs, the MMRs and EERs of MicPro are higher than Mcadam's and comparable with Voicemask's. Moreover, MicPro (T2) can also reach a high MMR of about 65% and EER of 20%, which is similar to (T1). We also present the ROC curves of ECAPA-TDNN in Fig. 11(c). In conclusion, MicPro can achieve a high enough EER compared with baselines in the digital domain.

6.2.2 Usability. **STOI.** For the use of virtual calls, our focus is on human perception. STOI is a widely-used objective metric for quantifying speech intelligibility. To obtain a more comprehensive assessment of STOI performance, we evaluate the STOI scores of MicPro and baselines across four datasets, as shown in Fig. 12(a). The results indicate that MicPro outperforms both McAdam and VoiceMask in VoxCeleb1 and AISHELL datasets while performing slightly worse than McAdam in LibriSpeech and VCTK datasets. We have also included standard error bars for the STOI scores, which show that both MicPro and McAdam have standard errors of less than 0.5 across all four datasets.

Latency. Latency is a critical factor that affects the user experience of virtual applications such as calls or meetings. Our research indicates that changing the G.729 codec has no significant impact on latency. The total latency comprises the algorithm delay, processing delay, and transmission delay [24]. The algorithm delay is a fixed value of 15ms, while the processing delay varies depending on the hardware used, and the transmission delay depends on network conditions. Regardless of the uncertain transmission delay, we measure the time required to encode audio using both the original G.729 codec and our modified version on the Respeaker. Four audio files with different durations are encoded 10 times repeatedly, and the mean values are presented in Tab. 3. The calculated latency l is expressed in milliseconds (ms) and can be determined as follows:

$$l = \frac{t_{enc} \cdot t_f}{t_{dur}} + 15\text{ms} \tag{25}$$

Where 15ms is the fixed algorithm delay, t_{enc} is the encoding time, t_{dur} is the duration of the audio, and t_f is the frame length (10ms



(a) STOI (b) Subjective Quality (c) WER

Figure 12: The usability performance of anonymization systems. (a): STOI score of three methods across four datasets. `MicPro` gets the highest STOI score in VoxCeleb1 and AISHELL. (b): subjective quality of three methods in terms of clearness, naturalness, similarity, and acceptability. `MicPro` performs similarly to McAdam and much higher than VoiceMask. (c) WER of three methods across three ASRs. `MicPro` performs the best in crdnn-rnn and wav2vec2 and keeps a comparable performance in transformer with McAdam and VoiceMask.

Datasets	EER (T1/T2)	MMR (T1/T2)	STOI (T1)	WER (T2)
VoxCeleb1 (E)	14.49%/-	76.5%/-	0.775 ± 0.036	-
VCTK (E)	12.56%/15.10%	42.62%/68.33%	0.792 ± 0.050	2.03%
AISHELL (C)	18.09%/21.00%	57.14%/57.29%	0.788 ± 0.046	5.58%

Table 4: The overall performance of `MicPro` in three different datasets. E: English, C: Chinese.

Method	EER (D/I)	MMR (D/I)	STOI (D/I)
Original	0.00%/1.54%	0.00%/0.00%	$1.0/1.0^{\dagger}$
`MicPro`	16.92%/25.60%	51.54%/87.69%	0.737 ± 0.041/0.727 ± 0.034
McAdam	12.31%/24.62%	60.00%/82.30%	0.819 ± 0.040/0.782 ± 0.027
VoiceMask	6.92%/9.99%	44.62%/58.46%	0.705 ± 0.044/0.704 ± 0.038

Table 5: Results of Direct and Indirect human speakers. D: direct, I: indirect. †: We use original human speakers audio as the reference for STOI.

for G.729). We compared the relative latency of our modified version to the original G.729 codec and found that the difference was negligible, with an average relative increase of only 0.01%.

Subjective quality. Besides the STOI, we conduct a user study to evaluate the subjective quality of `MicPro`. The subjective quality score represents the clearness, naturalness, similarity, and acceptability of anonymized audio rated by 40 participants. As shown in Fig. 12(a), `MicPro` achieves a comparable performance with McAdam while the score of VoiceMask is lower than others. However, we note that usability and anonymity are trade-offs, so comparing usability alone is meaningless. As shown in Fig. 11(b), the anonymity of Mcadam is the worst among the three systems, indicating that Mcadam compromises anonymity for better audibility. In summary, `MicPro` can well maintain both anonymity and audibility.

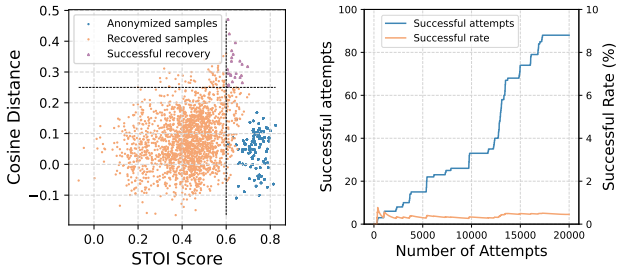
WER. Task 2 requires that the ASRs can correctively recognize the anonymized speech. Figure 12(c) shows the WERs of all three anonymization systems with three different ASRs. We observed that `MicPro` has the lowest WER for crdnn-rnn and wav2vec2 models. While for the transformer model, `MicPro` does not perform outstandingly, it still maintains an acceptable performance compared with the other two baselines. It means that our system can also meet the requirements of the traditional task.

6.3 Impact factors

6.3.1 Datasets. To figure out our system's generality, we experimented to compare the performance of `MicPro` in another English dataset (VCTK) and a Chinese dataset (AISHELL). The ASV model we use is a pretrained ECAPA-TDNN model, and the ASR models are a pretrained wav2vec2 model for VCTK and a transformer model for AISHELL. As Tab 4 shows, when we transfer our model to VCTK, another English dataset, `MicPro` degrades about 2% in EER and 10% in MMR. We assume that it is because VCTK is a high-quality audio dataset with a sample rate of 48kHz. As for AISHELL, `MicPro` can maintain high performance with an EER of about 19%, but the MMR drops at about 5%. Again, we assume that it is due to the difference in language. For the STOI and WER, `MicPro` keeps good performance in these two datasets. However, we believe these questions are existing engineering questions. Our system only suffers a slight influence with a decrease of EER below 2%, which illustrates that `MicPro` has a good generality in different datasets.

6.3.2 Voice Sources. From a more realistic perspective, the voice sources can be either direct human speakers or indirect human speakers speaking through loudspeakers in scenarios such as virtual calls. To evaluate the performance of `MicPro` in these cases, we conducted physical-domain experiments, following the setup presented in Fig. 10. Compared with the baselines, we list the anonymity and usability metrics in Tab. 5. The ASVs and ASRs for evaluation are ECAPA-TDNN and wav2wec, respectively. The results show that `MicPro` and McAdam yield a higher performance in anonymity but lower in usability due to the inevitable distortion in the physical domain, especially for indirect speakers. `MicPro` achieves better performance in all metrics compared with VoiceMask, and in EER and indirect MMR compared with McAdam.

6.3.3 Individualities. During the evaluation, we were concerned that the anonymity performance would show distinctiveness among different human speakers. It means some people's speech could be well anonymized while others could not. Therefore, we compute the standard deviation of MMRs of 40 different human speakers and get the result of 0.32. Compared with the std of 0.335 of Mcadam



(a) Scores (b) Successful attempts and rate

Figure 13: Results of the recovery attempts. (a): the cosine distances and STOI cores of anonymized and reversed samples. Only a few of samples marked with purple achieve a successful recovery. (b): the number of successful attempts and successful rate of audio reversing. We get 88 successful attempts out of 20,000. The successful rate is only 0.44%.

and 0.35 of VoiceMask, MicPro keeps an acceptable robustness to the impact of human speakers.

6.4 Security Analysis

In this subsection, we evaluate the security of MicPro against various threats, considering two types of attackers. (1) *Ignorant* attackers have no knowledge of the victim's use of an anonymization system and feed anonymized audio into ASVs to conduct *spoofing attacks* or *inference attacks*. (2) *Informed* attackers are aware of the victim's use of an anonymization system and attempt to recover audios or enroll anonymized audios into ASVs before conducting *spoofing attacks* or *inference attacks*.

6.4.1 Spoofing attack. Consider an *ignorant* attacker who attempts to use anonymized audio to bypass the victim's device identity verification process. The resistance of MicPro to such attacks can be observed in the MMR shown in Fig. 11(a), i.e., the probability that anonymized audio is rejected. The attack success rate can be calculated as 1-MMR. Notably, we observed that MicPro is most effective in X-Vector, achieving an attack success rate reduction of 1.95%. Even though the attack success rate is around 40% in ECAPA-TDNN, it still outperforms the baselines by around 20%.

An *informed* attacker possessing knowledge of our anonymization methods can recover the audio by a set of randomly guessed coefficients. An attacker can reverse the audio by the inverse transformation functions if she accidentally obtains coefficients, as the transformation function is both monotonic and invertible (see Sec. 5.2.2 for proofs). It is infeasible to perfectly recover the original audio due to the distortion introduced by the encoding process. Further more, The coefficients are distributed continuously in $[0, 2]^{1\times3} \subset \mathbb{R}^{1\times3}$. The probability of an attacker correctly guessing the coefficients is theoretically 0. However, to conduct a *spoofing attack*, the attacker does not need to recover the original audio perfectly. We therefore relax the definition of a successful recovery as one in which the voiceprint of the recovered audio matches the original one.

To assess the probability of an attacker successfully recovering an audio, we conducted a Monte-Carlo Sampling experiment using 100 successfully-anonymized audios from LibriSpeech-test-clean.

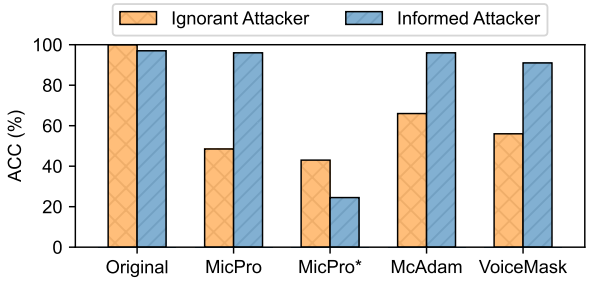


Figure 14: ACC of inference attack under ignorant and informed attackers. MicPro *: using random coefficients.

For each selected audio, we randomly generated 200 sets of coefficients and recovered the audio using the inversion functions of the transformation functions. We calculated the cosine distance between the recovered and original audios using the ECAPA-TDNN model, and marked the results with cosine distances larger than 0.25 and STOI scores greater than 0.6 (without loss of usability) as successfully-recovered samples.

Out of 20,000 attempts, we successfully recovered 88 audios, resulting in a success rate of 0.44%. Figure 13 shows that the vast majority of recovered audios did not match the voiceprint of the original and had severe degradation in speech quality. Figure 13(b) shows that the successful rate fluctuates around 0.44% when the number of attempts is large enough. Our experimental results demonstrate that our method is highly resistant to random recovery, with an expected probability of successful recovery of only 0.44%. If the attacker aims to have at least one successful recovery with a rate greater than p, the number of attempts N to make should satisfy $(1-0.0044)^N \leq 1-p$. For example, if $p = 0.99$, N should be at least 1045, and if $p = 0.95$, N should be at least 680. Even if the attacker generates a large number of recovery audios, she still doesn't know which attempt is correct.

6.4.2 Inference Attack. In this scenario, an attacker attempts to identify a victim's true identity by enrolling a pool of potential victims' speech to ASVs. An *ignorant* attacker does not know the audio has been anonymized and uses the original speech to enroll. In contrast, an *informed* attacker enrolls anonymized audios. The attacker then compares the victim's voiceprint with all enrollments, and the one with the highest ASV score is considered to be the victim's real identity. We conducted an inference attack on LibriSpeech-test-clean and calculated the inference accuracy (ACC), as shown in Fig. 14. Our results indicate that MicPro got an ACC of 48.5%, which is 17.5% lower than McAdam and 8.5% lower than VoiceMask.

However, all three methods suffer from resistance degradation when encountering an *informed* attacker, with an ACC of more than 90%. That's because for all three methods, the audio is converted in the same form with the same coefficients, and there is a high level of similarity between anonymized audio of the same speaker. To mitigate this threat, we can use MicPro with random available coefficients sourced from the feasible solutions. The results demonstrate that this random strategy dramatically reduces the threat of an *informed* attacker, reducing the ACC from 96% to 24.5%. In conclusion, our approach, MicPro, is more resistant to *inference attacks* under *informed* attackers.

7 DISCUSSION

MicPro is an SP-based method that uses a frame-by-frame approach for speech processing. While ML-based methods may be better suited for some tasks, SP-based methods are preferable for low-latency applications. However, they suffer from limited prior knowledge of voice features, making it challenging to find global optimal coefficients that balance anonymity and usability across different speakers. To address this, we can leverage the flexibility of CELP coding, which allows for modifying various aspects of voiceprints, such as pitch and excitation vectors. Future work can explore more applications of CELP coding for speech anonymization.

While MicPro is a promising prototype for privacy by design, our current implementation is limited to the G.729 codec, which has an 8kHz sampling rate and may not meet the requirements for high-quality audio transmission. However, we view this as an engineering challenge and believe that MicPro has broader potential beyond the scope of this paper. In our future work, we plan to adapt MicPro to 16kHz CELP codecs like G.722.2 (AMR-WB) and evaluate its performance in practical applications.

8 RELATED WORK

8.1 SP-based Anonymization

Signal processing (SP) methods can be used to modify speaker-specific features like pitch, formant positions, and speech rate. One way to do this is by using McAdams transformation to reposition the poles in the LPC filter, as demonstrated by Patino et al. [39]. Another SP method is vocal tract length normalization (VTLN) [17], which maps the frequency to another scale using a warping function. However, this method is reversible, making it vulnerable to de-anonymization attacks. Thus, Qian et al. [41, 42] use compound warping functions to improve the irreversibility.

One drawback of these SP-based methods is that they do not always separate the speaker-related and content-related features, resulting in speech quality degradation and unnatural-sounding speech that is incomprehensible. Therefore, it is generally believed that SP methods cannot guarantee both audibility and anonymity simultaneously. To address this challenge, Kai et al. [28] proposed a data-driven method that optimizes the parameters of SP methods, including VTLN, McAdams transformation, and other methods.

8.2 ML-based Anonymization

Compared with SP-based, ML-based anonymization is preferred for its better performance. Most anonymization techniques aim to fool ASV systems, which are often built using machine learning or deep learning methods. Therefore, exploring the vulnerabilities of ASVs is a useful method for anonymization.

Voice conversion (VC) and voice synthesis (VS) are two classical anonymization methods that work against ASVs by tampering with the original speaker's features. Justin et al. [27] transform the speaker feature to another. Bahmaninezhad et al. [6] map it to an average and anonymized feature. Fang et al. [18] randomly combine multiple speakers' vectors to access pseudo-speaker identities. While state-of-the-art VC/VS algorithms can achieve anonymity and maintain high-quality and naturalness of the audio, they are unsuitable for scenarios such as virtual calls.

Adversarial example (AE) is a popular method for exploiting neural network complexity and poses a threat to it. When used in anonymization systems, AE can protect privacy. Yi et al. [56] first proposed FAPG, which trains an AE generator and speakers' feature map to generate speaker-related adversarial examples to misguide traditional ASVs. V-Cloak, proposed by Deng et al. [15], achieves a transferable anonymizer. However, existing AE-based techniques lack interpretability and may not be suitable for lightweight microphone hardware with streaming input and output.

9 CONCLUSION

We propose MicPro, a privacy-preserving approach aimed at enabling hardware deployment for general use, especially in virtual calls and meetings. MicPro exploits the existing CELP codec, reducing the overhead of the algorithm while meeting the requirement of low latency. Therefore, it can be embedded and deployed in a microphone module in a lightweight manner. This paper presents the prototype of MicPro, that is, the modification of the speech is realized by modifying the signal processing flow of the CELP codec. In addition, this work uses a developed CELP codec and deploys it in a microphone to verify the feasibility of MicPro. Compared with the baseline, the flexible configuration of coefficients enables us to achieve a better trade-off between anonymity and usability. Future directions include exploring other privacy-preserving methods based on CELP and adapting MicPro for more applications.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable comments. We thank Xuancun Lu for helping with the physical-domain experiment. This work is supported by China NSFC Grant 62222114, 61925109, 62071428, and 62201503.

REFERENCES

[1] Bishnu S. Atal A. 2003. Speech Synthesis Based on Linear Prediction. *Encyclopedia of Physical Science and Technology (Third Edition)* (2003), 645–655.
[2] Amazon. 2014. Amazon Alexa. https://developer.amazon.com/alexa.
[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, and Bai et al. 2016. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*. PMLR, 173–182.
[4] Louie Andre. 2023. 53 Important Statistics About How Much Data Is Created Every Day. https://financesonline.com/how-much-data-is-created-every-day/.
[5] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc., 12449–12460.
[6] Fahimeh Bahmaninezhad, Chunlei Zhang, and John Hansen. 2018. Convolutional Neural Network Based Speaker De-Identification. In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*. 255–260.
[7] Adil Benyassine, Eyal Shlomot, H-Y Su, Dominique Massaloux, Claude Lamblin, and J-P Petit. 1997. ITU-T Recommendation G. 729 Annex B: a silence compression scheme for use with G. 729 optimized for V. 70 digital simultaneous voice and data applications. *IEEE Communications Magazine* 35, 9 (1997), 64–73.
[8] Bruno Bessette, Redwan Salami, Roch Lefebvre, Milan Jelinek, Jani Rotola-Pukkila, Janne Vainio, Hannu Mikkola, and Kari Jarvinen. 2002. The adaptive multirate wideband speech codec (AMR-WB). *IEEE transactions on speech and audio processing* 10, 8 (2002), 620–636.
[9] J. Blank and K. Deb. 2020. pymoo: Multi-Objective Optimization in Python. *IEEE Access* 8 (2020), 89497–89509.
[10] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*. IEEE, 1–5.

[11] Guangke Chen, Sen Chenb, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. 2021. Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems. In *2021 IEEE Symposium on Security and Privacy (SP)*. 694–711.

[12] Peterson-Barney database. 1995. https://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/speech/database/pb/.

[13] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

[14] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. 2010. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing* 19, 4 (2010), 788–798.

[15] Jiangyi Deng, Fei Teng, Yanjiao Chen, Xiaofu Chen, Zhaohui Wang, and Wenyuan Xu. 2023. V-Cloak: Intelligibility-, Naturalness- & Timbre-Preserving Real-Time Voice Anonymization. In *32nd USENIX Security Symposium (USENIX Security 23)*. 5181–5198.

[16] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. 2020. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Proceedings of Interspeech 2020*. 3830–3834.

[17] Ellen Eide and Herbert Gish. 1996. A parametric approach to vocal tract length normalization. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 1. IEEE, 346–348.

[18] Fuming Fang, Xin Wang, Junichi Yamagishi, Isao Echizen, Massimiliano Todisco, Nicholas Evans, and Jean-Francois Bonastre. 2019. Speaker anonymization using x-vector and neural waveform models. *arXiv preprint arXiv:1905.13561* (2019).

[19] V Muthu Ganesh and N Janukiruman. 2019. A survey of various effective Codec implementation methods with different real time applications. In *2019 international conference on communication and electronics systems (ICCES)*. IEEE, 1279–1283.

[20] Joaquín González-Rodríguez, Doroteo Torre Toledano, and Javier Ortega-García. 2008. Voice biometrics. In *Handbook of biometrics*. Springer, 151–170.

[21] Priyanka Gupta, Gauri P Prajapati, Shrishti Singh, Madhu R Kamble, and Hemant A Patil. 2020. Design of voice privacy system using linear prediction. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 543–549.

[22] Wenbin Huang, Wenjuan Tang, Hanyuan Chen, Hongbo Jiang, and Yaoxue Zhang. 2022. Unauthorized Microphone Access Restraint Based on User Behavior Perception in Mobile Devices. *IEEE Transactions on Mobile Computing* 01 (2022), 1–16.

[23] Wenbin Huang, Wenjuan Tang, Kuan Zhang, Haojin Zhu, and Yaoxue Zhang. 2022. Thwarting unauthorized voice eavesdropping via touch sensing in mobile systems. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 31–40.

[24] ITU-T. 2012. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP).

[25] ITU-T. 2021. G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP). https://www.itu.int/rec/T-REC-G.729.

[26] Sushil Jajodia and Henk CA van van Tilborg. 2011. *Encyclopedia of Cryptography and Security: L-Z*. Springer.

[27] Tadej Justin, Vitomir Štruc, Simon Dobrišek, Boštjan Vesnicer, Ivo Ipšić, and France Mihelič. 2015. Speaker de-identification using diphone recognition and speech synthesis. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Vol. 04. 1–7.

[28] Hiroto Kai, Shinnosuke Takamichi, Sayaka Shiota, and Hitoshi Kiya. 2021. Lightweight voice anonymization based on data-driven optimization of cascaded voice modification modules. In *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 560–566.

[29] Anssi Klapuri. 2006. Introduction to music transcription. In *Signal processing methods for music transcription*. Springer, 3–20.

[30] Jacob Leon Kröger, Otto Hans-Martin Lutz, and Philip Raschke. 2020. Privacy implications of voice and speech analysis–information disclosure by inference. *Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers 14* (2020), 242–258.

[31] Paul Lachat, Nadia Bennani, Veronika Rehn-Sonigo, Lionel Brunie, and Harald Kosch. 2022. Detecting Inference Attacks Involving Raw Sensor Data: A Case Study. *Sensors* 22, 21 (2022).

[32] Marianne Latinus and Pascal Belin. 2011. Human voice perception. *Current Biology* 21, 4 (2011), R143–R145.

[33] Jaemin Lim, Kiyeon Kim, Hyunwoo Yu, and Suk-Bok Lee. 2022. Overo: Sharing Private Audio Recordings. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 1933–1946.

[34] Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. 2021. Automatic speech recognition: a survey. *Multimedia Tools and Applications* 80 (2021), 9411–9457.

[35] Ian Vince McLoughlin. 2008. Line spectral pairs. *Signal processing* 88, 3 (2008), 448–467.

[36] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612* (2017).

[37] Toshiyuki Nomura and Masahiro Iwadare. 1999. Voice over IP systems with speech bitrate adaptation based on MPEG-4 wideband CELP. In *1999 IEEE Workshop on Speech Coding Proceedings. Model, Coders, and Error Criteria (Cat. No. 99EX351)*. IEEE, 132–134.

[38] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5206–5210.

[39] Jose Patino, Natalia Tomashenko, Massimiliano Todisco, Andreas Nautsch, and Nicholas Evans. 2021. Speaker Anonymisation Using the McAdams Coefficient. In *Interspeech 2021*. ISCA, 1099–1103.

[40] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.

[41] Jianwei Qian, Haohua Du, Jiahui Hou, Linlin Chen, Taeho Jung, and Xiang-Yang Li. 2019. Speech sanitizer: Speech content desensitization and voice anonymization. *IEEE Transactions on Dependable and Secure Computing* 18, 6 (2019), 2631–2642.

[42] Jianwei Qian, Haohua Du, Jiahui Hou, Linlin Chen, Taeho Jung, Xiang-Yang Li, Yu Wang, and Yanbo Deng. 2017. Voicemask: Anonymize and sanitize voice input on mobile devices. *arXiv preprint arXiv:1711.11460* (2017).

[43] Karthikeyan N Ramamurthy and Andreas S Spanias. 2010. MATLAB® software for the code excited linear prediction algorithm: The federal standard-1016. *Synthesis Lectures on Algorithms and Software in Engineering* 2, 1 (2010), 1–109.

[44] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. 2021. SpeechBrain: A General-Purpose Speech Toolkit. arXiv:2106.04624 [eess.AS] arXiv:2106.04624.

[45] Manfred Schroeder and B Atal. 1985. Code-excited linear prediction (CELP): High-quality speech at very low bit rates. In *ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 10. IEEE, 937–940.

[46] seeedstudio. 2021. ReSpeaker Core v2.0. https://wiki.seeedstudio.com/ReSpeaker_Core_v2.0/.

[47] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 5329–5333.

[48] F. Soong and B. Juang. 1984. Line spectrum pair (LSP) and speech data compression. *Proc. ICASSP, vol.1* 9 (1984), 37–40.

[49] Andreas S Spanias. 1994. Speech coding: A tutorial review. *Proc. IEEE* 82, 10 (1994), 1541–1582.

[50] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. 2011. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing* 19, 7 (2011), 2125–2136.

[51] Siri Team. 2018. Personalized Hey Siri. https://machinelearning.apple.com/research/personalized-hey-siri.

[52] Nuttakorn Thubthong and Boonserm Kijsirikul. 2001. Support vector machines for Thai phoneme recognition. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9, 06 (2001), 803–813.

[53] Tavish Vaidya and Micah Sherr. 2019. You Talk Too Much: Limiting Privacy Exposure Via Voice Input. In *2019 IEEE Security and Privacy Workshops (SPW)*. 84–91.

[54] VoicePrivacy2020. 2020. Voice Privacy Challenge 2020. https://github.com/Voice-Privacy-Challenge/Voice-Privacy-Challenge-2020/.

[55] Zhizheng Wu, Sheng Gao, Eng Siong Cling, and Haizhou Li. 2014. A study on replay attack and anti-spoofing for text-dependent speaker verification. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*. IEEE, 1–5.

[56] Yi Xie, Zhuohang Li, Cong Shi, Jian Liu, Yingying Chen, and Bo Yuan. 2021. Enabling fast and universal audio adversarial attack using generative model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 14129–14137.

[57] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. 2019. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit. University of Edinburgh.

[58] Young-Sun Yun, Jinman Jung, and Seongbae Eun. 2015. Voice Conversion Between Synthesized Bilingual Voices Using Line Spectral Frequencies. In *International Conference on Speech and Computer*. Springer, 463–471.

[59] yuunin. 2020. time-invariant-anonymization. https://github.com/yuunin/time-invariant-anonymization.

[60] Lei Zhang, Yan Meng, Jiahao Yu, Chong Xiang, Brandon Falk, and Haojin Zhu. 2020. Voiceprint mimicry attack towards speaker verification system in smart home. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 377–386.