USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

SpeechGuard: Recoverable and Customizable Speech Privacy Protection

Jingmiao Zhang, Suyuan Liu, Jiahui Hou, Zhiqiang Wang, Haikuo Yu, and Xiang-Yang Li, *University of Science and Technology of China*

https://www.usenix.org/conference/usenixsecurity25/presentation/zhang-jingmiao

This paper is included in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.

SPEECHGUARD: Recoverable and Customizable Speech Privacy Protection

Jingmiao Zhang, Suyuan Liu, Jiahui Hou*, Zhiqiang Wang, Haikuo Yu, and Xiang-Yang Li*
{jingmiao, lsysue}@mail.ustc.edu.cn, jhhou@ustc.edu.cn,
{zhiqiang.wang, yhk7786}@mail.ustc.edu.cn, xiangyangli@ustc.edu.cn
University of Science and Technology of China

Abstract

Uploading speech data to cloud servers poses privacy risks, making the protection of both acoustic and content privacy essential. Users often need the cloud to process non-sensitive information while protecting sensitive parts, with the ability to recover original data locally. However, achieving speech privacy protection that supports fine-grained customization and full recoverability remains a significant challenge. Existing methods often rely on irreversible or inflexible techniques, such as uniformly anonymizing the entire speech or replacing sensitive texts, making them inadequate for this purpose. We introduce SPEECHGUARD, a system that enables recoverable and customizable speech privacy protection by applying reversible protection methods and assigning private information to permission groups. We design a multi-parameter warping function with an inverse function for voice conversion to protect acoustic privacy. We also develop a mechanism for automatic or manual detection and encryption of sensitive texts to protect content privacy. By categorizing listeners into permission groups and assigning warping parameters and encryption keys, SPEECHGUARD enables different listeners to recover varying levels of acoustic and content information according to their permissions, ensuring personalized access to speech data. Experiments on three datasets show that SPEECHGUARD outperforms three baseline systems in anonymity, sensitive content confidentiality, and attack resistance. Moreover, it provides recoverable and customizable protection for acoustic and content privacy, allowing for tailored privacy definitions and protection strength.

1 Introduction

Speech data contains personal biometrics [1, 2] and speech habits [3, 4] that can uniquely identify an individual. Processing or storing speech data on cloud servers poses privacy risks, as it can be intercepted [5], misused [6], or shared with unauthorized third parties [7]. Within a single audio file, there may be sensitive and non-sensitive information. Although audio owners typically allow cloud servers to process non-sensitive information to enhance their experience, they are reluctant to expose their private information. They might want to share their real voice with friends but a protected version with strangers on social media, or restrict certain content for children while granting full access to adults [8]. These diverse needs make uniform protection impractical. Applying blanket protection would indiscriminately block access to all information, undermining the goal of selectively sharing information. At the same time, recovering private information may be necessary in certain cases. Audio owners do not want to lose any information stored in the cloud while protecting their privacy. Law enforcement agencies may require access to untampered recordings during investigations, as courts require evidence to be authentic and coherent [9]. This makes irreversible protection methods inadequate.

We categorize speech privacy into two types: *acoustic privacy*, which concerns the unique characteristics of a speaker's voice, and *content privacy*, which involves personal information such as names, phone numbers, addresses, etc. Previous works [10–13] address desensitization of both types of sensitive information, but none of them consider the needs of audio owners for recoverability and customizability. Our objective is to design a speech privacy protection system that protects acoustic and content privacy while allowing *recovery* and *customization*. This can be divided into the following three goals: (1) audio owners can precisely specify which acoustic and content information should be protected as private and how to balance privacy and speech quality; (2) listeners with different permissions can hear different private information; (3) protected speech can be recovered to its original form based on specific permissions.

To achieve these goals, the main challenge is that privacy protection, recoverability, and customizability are tightly coupled and cannot be handled separately. Solutions must be designed under the premise of recoverability and customizability, making existing methods insufficient. First, construct-

*Corresponding authors.

Table 1: SPEECHGUARD versus existing works.

Method	Content Privacy	Acoustic Privacy	User-agnostic*	Recoverability		Customization#
				Content	Acoustic	
McAdams [17]	✗	✓	✓	✗	✓	✗
Preech [10]	✓	✓	✗	✗	✗	✗
VoiceMask [11, 12]	✓	✓	✓	✗	✓	✗
Overo [13]	✓	✓	✓	✗	✓	✗
SPEECHGUARD	✓	✓	✓	✓	✓	✓

*: Whether the method does not require training for a new user, #: Whether audio owners can set personalized privacy permissions for different listeners so that they can access different private information.

ing a highly secure and recoverable protection scheme is difficult. Traditional speech protection methods render private information unrecoverable. For acoustic privacy, existing approaches primarily use speech synthesis [14–16] or voice conversion [11–13, 17] to anonymize the voice. However, speech synthesis results in the permanent loss of the original acoustic information, while voice conversion is vulnerable to attacks. For content privacy, replacing sensitive words with similar ones [10–12] or masking them [13] also makes the protected speech content unrecoverable. Second, customizing speech for each listener is challenging. Previous efforts ignore the possibility of allowing different listeners to access different private information within the same speech. The most straightforward solution is to create separate copies for each listener, but this significantly increases the cost of processing, communication, and storage. Our goal is to publish a single version while offering multiple personalized variants. This is not merely an access control problem, as it requires selective access to different private information within a single audio file. Traditional access control mechanisms, which work at the file level, are not capable of meeting this requirement.

In this paper, we make the first attempt to design a recoverable and customizable speech privacy protection and publishing system called SPEECHGUARD. SPEECHGUARD enables precise and flexible control over the protection of both acoustic and content features, as well as customizable levels of access for different listeners. A comparison between SPEECHGUARD and existing works is provided in Table 1. The key insight is that acoustic and content privacy can be finely divided and independently protected with reversible methods, making it possible to achieve privacy protection, recovery, and customization simultaneously. SPEECHGUARD provides fine-grained privacy protection from acoustic and content perspectives, identifying privacy and implementing protection operations at the frame level. By applying frame-level access control, SPEECHGUARD allows precise and flexible privacy protection. For acoustic privacy, we design a multi-parameter warping function for voice conversion. It distorts the spectrum of each frame with adjustable random parameters, allowing the strength of the privacy protection to be fine-tuned. This

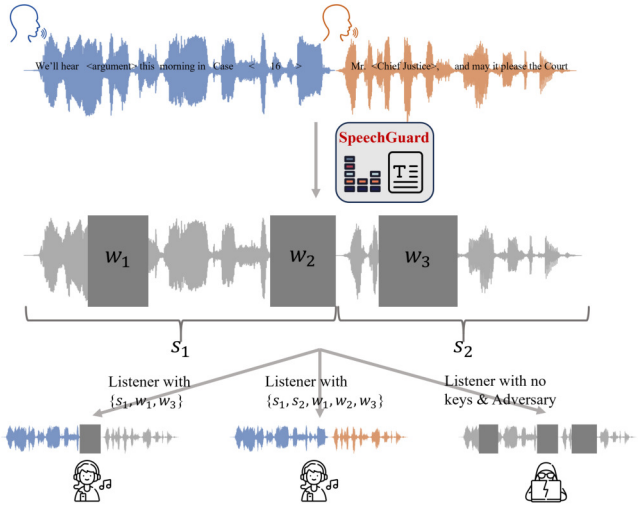


Figure 1: Basic functions of SPEECHGUARD. SPEECHGUARD protects both acoustic and content privacy of speech, allowing audio owners to define different levels of access to private information for listeners. The system accepts audio streams or recordings from multiple speakers, selects a parameter set S for acoustic privacy protection, generates a key set W for content privacy protection, and enforces access control over these parameters and keys. Listeners receive relevant parameters and keys based on their permissions, allowing them to access the protected information according to their authorization.

warping is reversible, and with knowledge of the parameters, the original voice can be recovered. For content privacy, sensitive content can be automatically or manually identified and protected by encrypting the corresponding frames. With the correct keys, the sensitive content can be recovered by decryption. When publishing audio, SPEECHGUARD supports customized privacy protection through access control on warping parameters and encryption keys. Authorized listeners access private information based on their designated authority using private keys. With full access to all parameters and keys, the protected audio can be completely recovered.

Figure 1 illustrates the basic functions of SPEECHGUARD. Furthermore, the system is robust to lossy audio compression and supports storage and transmission in the MPEG Audio Layer III (MP3) format.

We implement a fully functional prototype of SPEECHGUARD and evaluate it using state-of-the-art automatic speaker verification and speech recognition models on two public datasets and one volunteer dataset. Comparison with three baseline methods demonstrates that SPEECHGUARD offers the best anonymity and is the most resistant to attacks while maintaining comparable speech quality. The protection of sensitive content also surpasses that of the baselines and is the only one that allows recovery. A user study further shows that participants are satisfied with the system's usability, protection effectiveness, recovery performance, and overall performance. Comprehensive evaluation indicates that SPEECHGUARD allows authorized listeners to recover or partially recover original acoustic or content private information based on their permissions, while unauthorized listeners cannot.

In summary, this paper presents three main contributions:

- To the best of our knowledge, we propose the first recoverable and customizable speech protection system, SPEECHGUARD. It protects privacy while maintaining a certain level of speech quality and offers flexibility in adjusting protection strength and listener permissions. SPEECHGUARD allows a single version of audio to be customized for various listeners with different levels of access, and the original version can be recovered with specific listener permissions.
- We design a multi-parameter warping function for voice conversion to achieve better anonymity and robustness against attacks. This function is independent of the speaker and does not require pre-training.
- We conduct extensive experiments to verify the security, effectiveness, and efficiency of SPEECHGUARD and carry out a user study to confirm its practicality and applicability.

2 Background

2.1 Voice Conversion

The purpose of voice conversion is to disguise the speaker's voice without altering the linguistic content. One of the widely adopted voice conversion approaches is the frequency warping method [18] based on Vocal Tract Length Normalization (VTLN) [19, 20]. VTLN involves compensating for spectral variations caused by differences in vocal tract lengths, effectively adjusting the speech's spectral characteristics. This method is particularly suitable for real-time applications as it operates without pre-trained models. Additionally, voice conversion functions typically possess reversible properties. Knowing the conversion parameters allows for reversing the voice alteration [21, 22]. Building upon this foundation, designing acoustic privacy protection methods enables recoverability, but it also introduces security risks. Our work enhances security in this context.

2.2 Ciphertext-Policy Attribute-Based Encryption

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [23] is a public key cryptography technique designed for intricate access control on encrypted data. In CP-ABE, attributes describe a user's credentials. The data encryptor sets a policy that dictates who can decrypt it. Access control policies are defined using Boolean expressions involving these attributes, specifying the conditions required to decrypt and access the data. When data is encrypted, it is associated with a policy, and the encryption keys ensure that only users whose attributes meet the policy criteria can decrypt the data. CP-ABE is valuable in scenarios requiring flexible access control. Previous works use CP-ABE to protect sensitive files stored in the cloud or on IoT devices [24, 25]. However, these applications do not directly address fine-grained privacy protection within files. Our work integrates CP-ABE with fine-grained speech privacy protection, allowing listeners with different permissions to access various sensitive information within the same audio file.

2.3 Speech and Language Processing Tasks

Automatic Speaker Verification (ASV) aims to determine whether a given speech is spoken by a specific speaker. It consists of two phases: the registration phase and the inference phase. During the registration phase, the system collects speech samples from the target speaker and extracts voice embeddings that represent their unique voice characteristics. In the inference phase, the model compares the voice embeddings of the input speech with those of the registered speaker, producing a matching score. A higher matching score indicates a greater likelihood that the input speech originates from the target speaker [1, 2, 26].

Automatic Speech Recognition (ASR), also known as speech-to-text conversion, transforms spoken language into written text using a system comprising three key components: the acoustic model [27], lexicon model [28, 29], and language model [30, 31]. The acoustic model addresses the acoustic properties of speech by segmenting the audio signal into phonemes, the fundamental sound units of a language. Techniques such as Hidden Markov Models (HMMs) [32] and deep neural networks (DNNs) [33, 34] are used by the acoustic model to assess the probability of each phoneme or sub-phoneme given the speech input. The lexicon model contains information about the vocabulary of the language being

recognized, mapping words to their corresponding phonemic representations. Finally, the language model takes the recognized word sequence and assigns probabilities to different word sequences based on language patterns and context, thus helping to select the most probable transcription for the given speech.

Speaker Diarization (SD) is a task to label speech with classes corresponding to speaker identity, or simply put, to identify "who spoke when". This task involves several key steps. Firstly, speaker segmentation divides the audio stream into smaller segments based on discernible changes in speakers or sound sources, often relying on features like speaker characteristics, pitch, or pauses. Next, speaker clustering groups similar segments, associating them with the same speaker using clustering algorithms. The pivotal phase of speaker labeling follows, wherein each speaker cluster is assigned a specific identity or label, allowing for the identification of individual speakers or sound sources within the audio. Finally, speaker change detection plays a critical role in pinpointing moments in the audio where a speaker change occurs, facilitating precise segmentation and accurate labeling of different speakers or sources [35–37].

Named Entity Recognition (NER) focuses on identifying and classifying entities in text, such as people's names, places, organizations, dates, etc. Similar to ASV's speech embeddings, NER extracts embeddings that represent unique features of recognized named entities in speech. These embeddings capture distinctive characteristics that facilitate the effective identification and classification of entities. During the inference phase, the model compares these embeddings with predefined categories, generating a matching score to identify relevant entities in the input speech [38, 39].

3 Motivation

3.1 Use Cases

SPEECHGUARD is suitable for scenarios where recoverable privacy protection is needed or where single speech processing enables different versions to be presented to different listeners.

Audio recordings as evidence. Audio recordings are often used as evidence in legal cases and may undergo privacy processing before archiving. However, legal regulations mandate that recordings presented as trial evidence must possess clarity, authenticity, and coherence [9]. Voice anonymization can compromise clarity, and content desensitization may disrupt authenticity and coherence. SPEECHGUARD protects speech privacy while preserving the ability to recover the original version for legal purposes.

Posting audio recordings on social media. Audio recordings shared on social media can leak identity privacy and may require tailored presentation. SPEECHGUARD enables audio owners to apply privacy measures to both speech acoustics and content before sharing, allowing them to customize privacy levels for different audiences. For example, they can share their real voice with friends while protecting it from strangers, or adjust content for different age groups.

Eliminating specific speakers in meetings. In network conferences, recording requires consent from all participants [40–42]. Some may prefer their utterances not to be recorded. SPEECHGUARD allows for masking the speech of these participants while preserving the recordings of others.

3.2 Threat Model

We consider a scenario involving an audio owner and several listeners categorized based on their permission levels: L_1 listeners can access all private information, L_2 listeners can access partial private information, and L_3 listeners cannot access any private information. The audio owner publicly publishes privacy-processed audio and distributes keys to L_1 and L_2. L_1 listeners can recover all private information with their private keys, effectively having permissions equivalent to those of the audio owner. L_2 listeners can only recover parts of the private information, with different L_2 listeners recovering different private information depending on their permissions. L_3 listeners do not possess private keys and cannot recover any private information.

In our threat model, we assume the adversary has the following capabilities: (1) access to the privacy-processed audio published by the audio owner; (2) guess the parameters and keys used during the protection process; (3) use existing ASV algorithms to infer the speaker's identity; (4) use existing ASR algorithms to infer protected semantic content. While previous speech protection systems typically consider L_3 as the adversary, in our scenario both L_2 and L_3 may act as adversaries, as L_2 listeners might be curious about private information beyond their permissions.

Furthermore, the audio owner may publish audio through a public cloud. We assume that such transmission channels and storage devices are honest but curious, meaning they perform their duties without tampering with the audio but may seek to learn private information. The audio owner authenticates listeners during the key distribution process, but the specific authentication protocol is beyond the scope of this paper.

3.3 Problem Statement

Our goal is to protect privacy while maintaining speech quality and allowing the privacy-processed speech to be fully or partially recovered under certain permissions. We define the problem as follows:

Given a speech sample x, the audio owner performs acoustic and content privacy protection, resulting in x_p. The listener then recovers x_r from x_p.

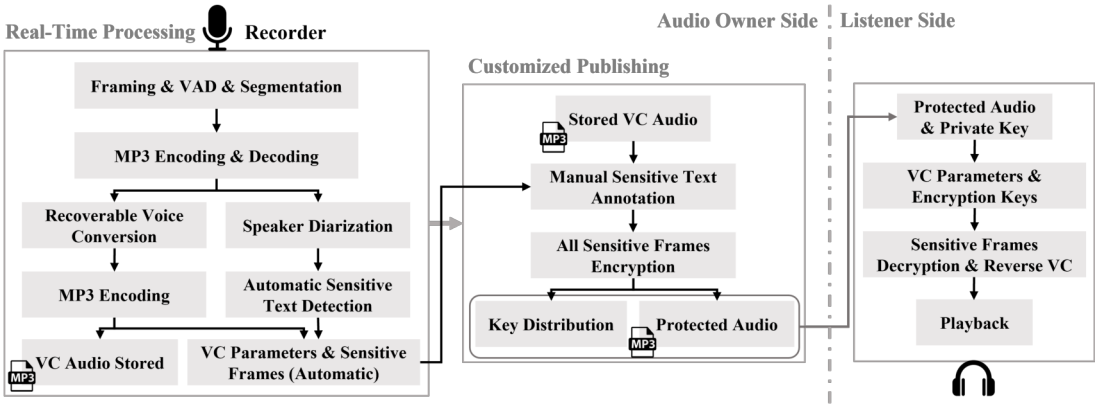


Figure 2: High-level overview of SPEECHGUARD.

Problem 1 (Trade-off between privacy and speech quality) *Our system aims to reduce the ASV accuracy of x_p, and reduce the ASR accuracy related to sensitive content in x_p, while maintaining the ASR accuracy related to non-sensitive content in x_p.*

Problem 2 (Recoverability) *Within the listener's permissions, our system aims to improve the ASV accuracy of x_r to approach that of x as closely as possible, and improve the ASR accuracy related to sensitive content in x_r to approach that of x as closely as possible.*

Problem 3 (Customizability) *The system allows the audio owner to define privacy settings by specifying the transformation from x to x_p, determining which speakers to anonymize, which texts to mark as sensitive, and how to balance privacy and speech quality. Additionally, it enables the owner to assign permissions to listeners, such that each listener recovers x_r based on their specific permissions, allowing fine-grained control over access to acoustic and content privacy.*

4 SPEECHGUARD: Design Details

We propose SPEECHGUARD, a recoverable and customizable speech protection system, as shown in Figure 2. This comprehensive solution involves both the audio owner and listener sides. The audio owner records real-world audio, which is processed in real time by the system to provide initial protection measures, including voice conversion and automatic sensitive text detection. These processed results are stored locally on the recording device. During the publishing phase, audio owners can freely adjust the protection by adding or removing sensitive texts. The system then encrypts all designated sensitive texts, and the warping parameters and encryption keys are distributed to listeners based on their permissions. Listeners equipped with their respective private keys can decrypt sensitive information they are authorized to access, thereby recovering relevant acoustic and content privacy according to their permissions. To optimize storage and transmission efficiency, SPEECHGUARD supports compressing the original recordings and the protected versions into the MP3 encoding format.

4.1 Data Preprocessing

SPEECHGUARD accepts real-time audio streams or audio recordings as input and operates on a frame-based methodology, with each frame spanning 36 milliseconds in this paper. This approach provides precise control over speech acoustic privacy and content privacy, enabling independent privacy protection at the frame level.

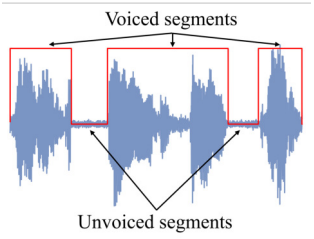


Figure 3: Dynamic threshold VAD and segmentation.

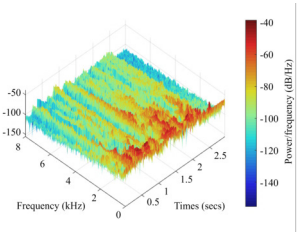


Figure 4: Short time Fourier transform (STFT).

To address potential environmental noise, we employ a dynamic threshold voice activity detection (VAD) technique [43] to partition the audio stream into voiced segments and unvoiced segments, as shown in Figure 3. Assuming there are no multiple people speaking simultaneously, this method ensures that each segment contains speech from a single individual or no speech. We apply short-time Fourier transform (STFT) on each frame, as shown in Figure 4, and calculate the threshold $\Phi(k)$ to determine whether each frame is a voiced frame or an unvoiced frame using the formula:

$$\Phi(k) = a_t \left(\max(|X_k(e^{j\omega})|) + \delta\right) + \sum_{i=1}^{T} a_i \Phi(k-i) \quad (1)$$

where $X_k(e^{j\omega})$ represents the STFT for the k^{th} frame, a_t is a real coefficient that weights the actual value of the maximum peak in the STFT, δ is a band guard ensuring that the new threshold remains above the noise floor, T denotes the prediction order, indicating the use of the first T thresholds to predict the expected value of $\Phi(k)$, a_i signifies the prediction coefficient, assigning weights to previous threshold values. Frames are classified as unvoiced if $\max(|X_k(e^{j\omega})|) < \Phi(k)$, or as voiced if $\max(|X_k(e^{j\omega})|) \geq \Phi(k)$. These continuous voiced frames and unvoiced frames collectively constitute voiced segments and unvoiced segments respectively. We set $T = 1$, $a_t = 0.7$, $a_i = 0.25$, $\Phi(-1) = \max(|X_k(e^{j\omega})|) + \delta$, $\delta = 1.99$ in this paper.

4.2 Voice Anonymization

We develop a VTLN-based voice anonymization method to protect acoustic privacy while maintaining speech quality. VTLN-based methods use an invertible warping function for voice conversion, allowing the recovery of the original state. However, the existing VTLN-based methods are vulnerable to deanonymization attacks. Therefore, we design a multi-parameter warping function that improves security by increasing the number of parameters and introduce a method for selecting the appropriate parameter range.

4.2.1 Basic Warping Function and Its Vulnerability

One of the most commonly used warping functions is the symmetric piecewise linear function with two segments [18], defined as:

$$p(\omega,\alpha) = \begin{cases} \alpha\omega, & \text{if } \omega \leq \omega_0 \\ \alpha\omega_0 + \frac{\pi-\alpha\omega_0}{\pi-\omega_0}(\omega-\omega_0), & \text{if } \omega > \omega_0 \end{cases} \quad (2)$$

where ω represents the normalized frequency in the range $[0,\pi]$, α is the warping parameter, and ω_0 is the turning point in the piesewise linear function. When $\alpha \leq 1$, $\omega_0 = \frac{7}{8}\pi$; for $\alpha > 1$, $\omega_0 = \frac{7}{8\alpha}\pi$. Given an audio signal and a warping parameter α, the piecewise linear function calculates a new frequency, resulting in a distorted audio output. When $\alpha > 1$, the new frequency surpasses the original frequency, resulting in a sharper-sounding output. Conversely, when $0 < \alpha < 1$, the new frequency is lower than the original frequency, resulting in a deeper-sounding output. When $\alpha = 1$, the original frequency remains unchanged, producing no distortion.

Warping functions are invertible, allowing voice conversion to be reversed. For instance, with the piecewise linear function, knowing the warping parameter α allows us to use its inverse function $p^{-1}(\omega,\alpha)$ to recover the protected speech to its original form. However, this feature also introduces a vulnerability, as a single parameter can be relatively easy for attackers to guess. Another potential threat is the reducing attack, initially discussed in [11]. This attack occurs when the attacker inaccurately guesses the warping parameter but makes an approximate estimation. Consequently, using this estimated parameter to reverse the voice conversion produces audio that closely resembles the original, even if the parameter is not precisely determined.

To address vulnerabilities, certain studies [11, 13] suggest dual-parameter voice conversion. However, they still have shortcomings, such as a fixed number of parameters, a constant turning point, and a single direction of spectrum distortion.

4.2.2 Multi-Parameter Warping Function

We propose a multi-parameter voice conversion method to improve security by increasing the number of parameters. This method randomly determines the number of parameters and generates them within an appropriate range for voice conversion. Assuming the generation parameters $\alpha = \{\alpha_1,\ldots,\alpha_n\}$ and $\beta = \{\beta_1,\ldots,\beta_n\}$, the multi-parameter warping function is defined as:

$$m(\omega,\alpha,\beta) = \frac{\beta_i-\beta_{i-1}}{\alpha_i-\alpha_{i-1}}(\omega-\alpha_i)+\beta_i, \text{if } \alpha_{i-1} \leq \omega \leq \alpha_i \quad (3)$$

where α,β are warping parameters. We predefine $\alpha_0 = \beta_0 = 0$, $\alpha_{n+1} = \beta_{n+1} = \pi$, with $\alpha_0 < \alpha_1 < \cdots < \alpha_{n+1}$, $\beta_0 < \beta_1 < \cdots < \beta_{n+1}$ and $1 \leq i \leq n+1$. The number of parameter pairs, denoted as n, and their values are randomly selected. The n turning points are modulated by α, while the amplitude of spectrum distortion is governed by β. Specific values of these parameters can induce spectrum distortion in various directions. Comparisons among the SPEECHGUARD warping function (multi-parameter), the piecewise linear function (single-parameter), and the Overo warping function (dual-parameter) [13] are shown in Figure 5. An example of the spectrograms before and after distortion using the multi-parameter warping function is presented in Figure 6.

The inverse function of the multi-parameter warping function $m(\omega,\alpha,\beta)$ is denoted as m^{-1}, where $m^{-1}(\omega,\alpha,\beta) = m(\omega,\beta,\alpha)$. If the warping parameters are known, applying m^{-1} to the protected voice can reverse the voice conversion.

Assuming there are a total of v segments after preprocessing, each segment is assigned a unique set of parameters $s = \{\alpha_1,\ldots,\alpha_n,\beta_1,\ldots,\beta_n\}$, resulting in each segment utilizing a distinct $2n$ parameters for voice conversion. All warping parameters are collectively denoted as $S = \{s_1,\ldots,s_v\}$.

Appropriate range for warping parameters. We follow the definition of distortion strength in [11] to quantify the intensity of voice conversion. The distortion strength, denoted by $dist$, for a warping function $\mathcal{F}(\omega,\mathcal{A})$ is defined as the area between the curves of the function and the identity function.

$$dist_{\mathcal{F}}(\mathcal{A}) = \int_0^{\pi} |\mathcal{F}(\omega,\mathcal{A}) - \omega| \quad (4)$$

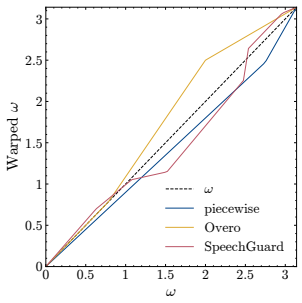


Figure 5: Warping functions. SPEECHGUARD has a random number of parameters, multiple turning points, and variable warping directions.

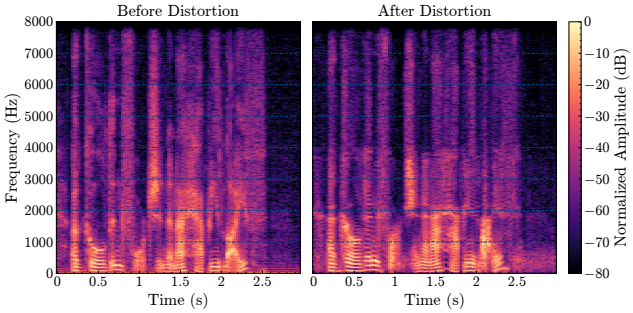


Figure 6: Spectrograms before and after distortion using multi-parameter warping function. This function stretches some frequencies while compressing others.

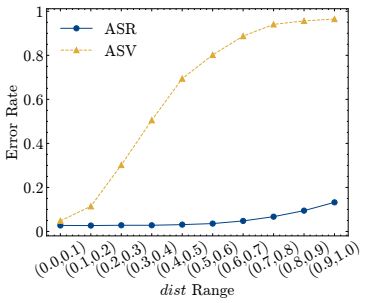


Figure 7: ASR and ASV error rates vary with *dist*. As the *dist* value increases, the ASR error rate rises gradually, while the ASV error rate climbs rapidly.

where $\mathcal{A}$ represents the warping parameter(s). A higher value of *dist* indicates a more significant distortion in the output speech.

We transform the exploration of the appropriate range of warping parameters into determining distortion strengths that balance anonymity and speech quality. To illustrate our method, we use the Librispeech dev-clean dataset [44], focusing on the case $n = 1$ where the multi-parameter warping function varies only at a single pair of parameters $\{\alpha, \beta\}$. On the original dataset, the ASR error rate with the SpeechBrain Transformer model [45] is 2.08%, and the ASV error rate with the NVIDIA TitaNet-Large model [37] is 2.52%. Voice conversion increases error rates in both ASR and ASV, with ASR error rates rising gradually and ASV error rates spiking as distortion strength increases, as shown in Figure 7. This significant discrepancy provides an opportunity to find an appropriate range of distortion strengths to suppress the possibility of speaker verification while maintaining speech recognition capabilities. The patterns for $n = 2, 3, \ldots, 10$ are similar to those for $n = 1$, as shown in Table 2. Selecting the appropriate range involves a trade-off between privacy and speech quality. A higher ASV error rate enhances privacy, while a lower ASR error rate improves speech quality. We limit the distortion strength to (0.5, 0.6), where the ASV error rate reaches 80.18% and the ASR error rate is 3.58%. Audio owners can relax or tighten their *dist* values based on their specific privacy preferences.

4.3 Sensitive Text Encryption

SPEECHGUARD protects speech content privacy by encrypting sensitive texts in the speech, allowing for recovery when needed. Our content unit is the word, while phrases and sentences are masked by encrypting all frames from the first frame of the starting word to the last frame of the ending word. We first identify and precisely locate the sensitive texts, then encrypt only those portions of the audio. Additionally,

Table 2: EER and WER vary with the *dist* range under different n values. The error rate for ASV is twice the EER, and the error rate for ASR is the WER.

	n	(0, 0.1)	(0.1, 0.2)	(0.2, 0.3)	(0.3, 0.4)	(0.4, 0.5)	(0.5, 0.6)	(0.6, 0.7)	(0.7, 0.8)	(0.8, 0.9)	(0.9, 1)
EER (%)	1	2.40	5.72	15.10	25.26	34.71	40.09	44.36	47.03	47.82	48.24
	2	2.74	7.42	15.90	23.58	31.39	38.04	40.75	43.51	43.83	44.77
	3	2.77	6.86	14.90	23.19	31.66	36.95	39.37	41.44	43.54	44.78
	4	2.81	7.50	14.90	23.20	30.42	35.75	39.54	41.63	42.42	43.93
	5	2.77	6.14	14.35	22.81	29.71	35.96	39.82	41.05	42.99	43.58
	6	2.73	6.51	14.28	22.37	29.69	35.18	38.01	41.41	42.19	43.31
	7	2.73	6.32	13.61	22.88	29.08	34.69	37.60	40.26	41.91	42.26
	8	2.64	6.33	13.39	21.92	30.01	33.96	37.33	39.49	40.45	42.04
	9	2.57	6.51	13.01	20.90	29.28	33.54	36.76	39.07	40.36	42.07
	10	2.66	6.16	13.22	21.00	27.58	32.82	36.49	38.35	39.32	40.31
WER (%)	1	2.74	2.68	2.82	2.81	3.11	3.58	4.78	6.72	9.46	13.22
	2	2.74	2.97	2.88	3.11	4.04	5.88	7.49	11.26	14.56	18.02
	3	2.73	2.84	2.91	3.30	3.89	5.27	8.17	11.11	13.73	20.29
	4	2.79	2.82	2.94	3.34	3.84	5.56	7.80	10.23	15.12	18.32
	5	2.78	2.77	2.96	3.28	3.94	5.27	7.97	10.27	15.11	18.28
	6	2.72	2.79	2.91	3.15	3.85	5.63	7.52	11.72	13.75	17.30
	7	2.75	2.80	2.99	3.25	4.05	5.07	7.79	9.98	13.22	16.49
	8	2.77	2.85	2.97	3.19	4.02	5.45	7.38	9.35	12.11	14.86
	9	2.74	2.83	3.02	3.30	3.92	5.03	6.85	9.11	10.88	13.72
	10	2.76	2.82	2.98	3.29	3.99	4.66	6.40	9.08	10.68	12.55

ensuring correct playback of unencrypted sections is crucial, especially with lossy compression formats. We adapt SPEECHGUARD to the MP3 format.

4.3.1 Automatic Sensitive Text Detection

Automatic sensitive text detection is designed to quickly identify predefined sensitive words and patterns. These predefined sensitive texts are based on legal regulations, such as the General Data Protection Regulation (GDPR), which protects the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data used to uniquely identify a natural person, data concerning health, or data concerning a natural person's sex life or sexual orientation [46]. Additionally, our system provides audio owners with the flexibility to create customized sensitive word lists, as they may have individual preferences regarding what they consider sensitive words. In cases where audio owners cannot accurately predefine words that may appear in the speech, such as phone numbers and home ad-

dresses, typically in the form of phrases, they can describe their patterns. For such cases, we use NER to target predefined patterns. Our system also supports setting different sensitive texts for different listeners to meet diverse privacy requirements. The sensitive words and patterns are set before the audio is recorded and stored locally in hash tables, and they are not shared with any other party.

We perform SD on the speech to obtain the transcript and the speaker of each segment. Next, we input the speech and transcript into a forced aligner [47, 48] to perform speech-text alignment. These steps allow us to obtain the transcribed text from the speech and the timing information indicating the start and end time of each word in the speech relative to the recording's start time. Finally, by searching the aligner's output for the sensitive words and patterns contained in the predefined lists, we can quickly locate the exact position of the predefined sensitive texts in the speech.

4.3.2 Manual Sensitive Text Annotation

Why does SPEECHGUARD provide a manual annotation option? The main reason we provide a manual annotation option is that it is difficult for audio owners to anticipate all sensitive texts before recording. They may have temporary privacy requirements for adding or deleting automatically detected sensitive texts after recording. They may also want to encrypt entire sentences or paragraphs. Manual annotation compensates for this limitation. Additionally, the NER and SD models used in SPEECHGUARD must operate offline, as online speech models may pose privacy risks. However, offline speech models often make errors, as shown in Table 3, which details the error rates of the best NER and SD models across three datasets. NER may misclassify recognition types, and SD may missegment speaker transition points. Even minor errors, like missing a sensitive word or misclassifying a speaker, can undermine speech privacy, making manual correction essential. Our survey of 30 volunteers shows that while 18 prefer predefined sensitive words and 27 favor predefined patterns, 15 choose manual annotation. This also illustrates the necessity of manual annotation.

[11] is a known work that relies solely on automatic detection, collecting a list of sensitive words and using an adaptive keyword spotting mechanism based on dynamic time warping (DTW). However, this approach does not accommodate users who wish to adjust sensitive texts after recording. Additionally, the time complexity of this method is related to the number of predefined sensitive words. The more sensitive words there are, the slower the algorithm runs, which can make this step a time bottleneck for the system.

Discussion. Manual annotation is intended to provide audio owners with an option to change privacy requirements and correct automatic detection errors after the audio is recorded. However, if audio owners believe automatic detection is sufficient, they can choose not to use manual annotation.

Table 3: Error rates of offline speech models.

Dataset	NER (%)	SD (%)		
	BERT [49]	wav2vec 2.0 [50]	HuBERT [51]	WavLM [52]
Clean 1	5.00	6.09	5.93	4.04
Clean 2	4.68	5.60	5.84	3.48
Noisy	10.31	16.45	15.19	10.41

We use a symmetric encryption algorithm to encrypt all sensitive texts. For words, directly encrypt all frames from the start frame to the end frame. When it comes to phrases and sentences, we encrypt all frames from the first frame of the starting word to the last frame of the ending word. Assume that there are a total of c encryption objects, each assigned a key w. All encryption keys are collectively denoted as $W = \{w_1, \ldots, w_c\}$.

4.3.3 Adaptation to Lossy Compression MP3

For audio files in uncompressed or lossless compression formats, sensitive texts are directly encrypted after detection. However, for audio in lossy compression formats, the encoding format must be adapted to avoid playback errors. To facilitate storage and transmission, SPEECHGUARD employs MP3 encoding. MP3 files consist of a tag frame and several data frames. The tag frame contains metadata information about the audio file and indicates the tag size. Each data frame comprises a frame header and compressed sound data. The frame header records information such as the version number, bit rate, sampling rate, etc., from which the frame size and frame duration can be calculated. Tag frames and data frame headers are excluded from encryption. Assuming that the starting time of a sensitive text is t_s and the ending time is t_e, the frame where the sensitive text starts (f_s) and ends (f_e) are calculated as follows:

$$f_s = \lfloor \frac{t_s}{T_d} \rfloor, \quad f_e = \lceil \frac{t_e}{T_d} \rceil \tag{5}$$

where T_d is the frame duration. Frame f in the range $f_s \leq f \leq f_e$, except for the frame headers are encrypted. Encrypting sensitive texts does not impact the playback of the unencrypted portion, but the encrypted segment may exhibit a harsher sound. To address this, we develop a simple decoder. If the listener has legal permission for the sensitive texts, the decoder will decrypt them. Otherwise, frames associated with the sensitive texts will be substituted with silent frames to ensure good listening quality.

4.4 Authorization and Privacy Recovery

Our system protects privacy information at the frame level, with operations on each privacy frame being atomic and reversible. This allows the audio owner to display different

privacy information to different listeners. We establish access control on warping parameters $S = \{s_1, \ldots, s_v\}$ and encryption keys $W = \{w_1, \ldots, w_c\}$ to achieve fine-grained permission allocation for acoustic privacy and content privacy. By knowing the warping parameters s_i, $1 \leq i \leq v$, listeners can reverse the voice conversion of a segment. Similarly, with the correct encryption key w_j, $1 \leq j \leq c$, listeners can decrypt a sensitive text. Listeners are divided into permission groups, and the audio owner can assign different subsets of S and W to these groups. The audio owner can predefine these permission groups or modify them at the time of publishing, ensuring flexible control over what private information each group can recover.

To distribute warping parameters and encryption keys securely, we employ CP-ABE [23]. We treat the warping parameters and encryption keys corresponding to the same permission group as messages that need to be transmitted and embed the access structure into them. In this way, only the listeners of this permission group can recover the corresponding information to its original state before protection. If a permission group is allowed to access the complete set of warping parameters S and encryption keys W, it can reverse all voice conversions and sensitive text encryptions, thereby obtaining the original version of the speech. Let the permission group set be $G = \{G_1, \ldots, G_g\}$, where g is the number of permission groups. Let the audible privacy scope of permission group G_i, $1 \leq i \leq g$, be $M_i = \{S_i, W_i\}$, where $S_i \subset S$ and $W_i \subset W$. Its permission structure is $\mathbb{A}_i$, and its attribute set is Ω_i. The permission control scheme can be described as follows:

Setup. The setup algorithm takes no input other than the implicit security parameter. It outputs the public key PK_i and a master key MK_i.

Encrypt(PK_i, M_i, $\mathbb{A}_i$). The encryption algorithm takes as input the public key PK_i, the message which is the audible privacy scope M_i of G_i, and the access structure $\mathbb{A}_i$. It outputs the ciphertext CT_i.

KeyGen(MK_i, Ω_i). The key generation algorithm takes as input the master key MK_i and a set of attributes Ω_i that describe the key. It outputs a private key SK_i.

Decrypt(PK_i, CT_i, SK_i). The decryption algorithm takes as input the public key PK_i, a ciphertext CT_i which contains an access policy $\mathbb{A}_i$, and a private key SK_i for a set Ω_i of attributes. If the set Ω_i of attributes satisfies the access structure $\mathbb{A}_i$, then the algorithm will decrypt the ciphertext and return the message M_i.

5 Evaluation

5.1 Experiment Setup

Prototype. We randomly set the number of warping parameter pairs n for voice conversion to an integer within the range [1, 10], and restrict the distortion strength *dist* to the range (0.5, 0.6). We use pretrained models for automatic sensitive text detection: WavLM [52] for SD, BERT [49] for NER, and MFA [47] for forced alignment. Inference is executed on a P100 GPU with 16 GB of memory. For sensitive text encryption, we employ the AES-CTR algorithm [53], with keys generated randomly.

Dataset. We conduct experiments on three datasets. The LibriSpeech test-clean dataset [44], utilized as the clean dialogue corpus, consists of 2620 clear utterances read by 40 native American English speakers without background noise, totaling 5.4 hours. The CSTR VCTK Corpus [54] comprises speech data from 109 native English speakers with various accents, each reading approximately 400 utterances. Additionally, we enlist 18 volunteers to record a dataset, featuring 8 native English speakers and 10 non-native English speakers. Each volunteer read in the lab office for around 10 minutes, resulting in a total of 1300 utterances. The recording environment includes ambient noise from desktop computers, keyboard typing, and air conditioning. Table 4 provides the dataset statistics. All audio files are converted to 16 kHz 16-bit PCM encoded mono WAV format.

Table 4: Dataset statistics.

Dataset	#Speakers	#Utterances	Hours	Noisy	Accents
LS test-clean	40	2620	5.4	No	American
VCTK	109	8163	44.0	No	Various
Volunteers	18	1300	3.0	Yes	Various

We start by evaluating the performance of acoustic and content privacy protection separately. Next, we combine these methods, report the overall performance, and conclude with a user study to assess the manual aspects and gather user feedback. To assess voice anonymization, we use the state-of-the-art ECAPA-TDNN model [55] based on x-vector [56] to extract speaker embeddings, measure their similarity with cosine distance, and obtain ASV results. Additionally, we employ the state-of-the-art Conformer model [57] for ASR. For evaluating speech content desensitization, we check if predefined sensitive texts remain recognizable after automatic detection and encryption. Note that no training is needed throughout the SPEECHGUARD system. The ASV model used for evaluation is trained on the Voxceleb1 [58] and Voxceleb2 [59] training data, and the ASR model is trained on the LibriSpeech 960-hour training dataset [44].

Metrics. We evaluate SPEECHGUARD using 4 metrics in terms of privacy, speech quality and real-time performance.

- Equal Error Rate (EER), the point where the False Acceptance Rate (FAR) and False Rejection Rate (FRR) are equal. It is a commonly used metric in ASV systems to measure the anonymity of speech. A higher EER implies

lower verification accuracy, indicating stronger privacy. Notably, an EER of 0.5 signifies the optimal privacy level. As the EER approaches 0.5, speaker recognizability is almost lost.

- Word Error Rate (WER), measures the percentage of words in the system's output that differ from the ground truth transcript in ASR systems. WER is calculated by considering substitutions, insertions, and deletions in the recognized output compared to the ground truth. A lower WER reflects fewer errors in speech recognition, indicating higher speech quality.

- False Negative Rate (FNR), the proportion of sensitive words that the ASR model fails to correctly identify relative to the total number of sensitive words. FNR is employed to evaluate the protection of content privacy. A higher FNR, after encrypting sensitive words, indicates that more sensitive words cannot be detected, thereby enhancing privacy protection.

- Real-time Coefficient (RTC), calculated as the ratio of the time spent processing the speech to the duration of the original speech. RTC serves as a metric to evaluate the efficiency of SPEECHGUARD. The lower the RTC, the better the real-time performance.

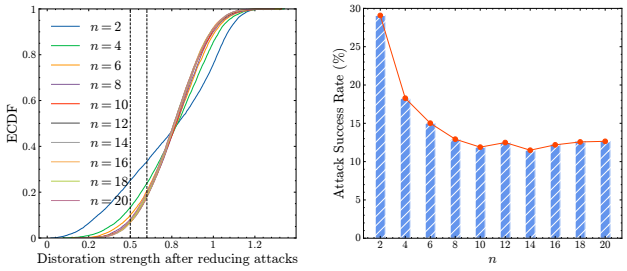
5.2 Acoustic Privacy

The cornerstone of acoustic privacy protection is the utilization of the multi-parameter warping function for voice conversion. We evaluate its performance from security, voice anonymity, speech quality and real-time performance perspectives.

Security against deanonymization attacks. As the multi-parameter warping function has an inverse function, an attacker could potentially reverse the voice conversion by knowing all the warping parameters. However, guessing all the parameters is challenging. Assuming the number of warping parameter pairs n is an integer that can take N values, the attacker must correctly guess n (with a probability of $\frac{1}{N}$) and then accurately guess $2n$ floating-point numbers within the range $(0, \pi)$. Given that the speech consists of v segments, each using new parameters, it would require v consecutive correct guesses for each segment to reverse the voice conversion entirely. This is nearly impossible.

We evaluate the security against reducing attacks through simulation experiments and real attacks on three datasets. We assume that the attacker knows n is an integer in [1, 10] and that the distortion strength *dist* is in (0.5, 0.6), but does not know the specific values of n and the warping parameters. Through 10,000 reducing attacks, the attacker uses randomly generated parameters within the appropriate range to calculate the inverse function of the multi-parameter warping function on the spectrum after voice conversion, attempting to recover the original spectrum. In only 15.87% of cases, the distortion strength after the attack is smaller than the distortion strength after voice conversion. In 84.13% of cases, the attack does not make the speech closer to the original but distorts it further. Moreover, the attacker cannot distinguish between successful and failed attempts.

We also explore the impact of the value of n on the success rate of the reducing attack. Figure 8(a) shows the empirical cumulative distribution function (ECDF) curve of 10,000 reducing attacks when n takes different fixed values, and Figure 8(b) shows the corresponding attack success rates. The results indicate that, in general, the more parameters there are, the better the resistance against reducing attacks. This supports our approach of designing a multi-parameter warping function to increase the number of parameters, thereby resisting attacker guessing. When $n > 10$, as n increases, the rate at which the attack success rate decreases slows down. On the other hand, Figure 9 shows that as n increases, the EER under the same *dist* range decreases, making anonymity worse. To balance anonymity and robustness against attacks, we set n to be within the range of [1, 10] as the default value in the following experiments.



(a) (b)

Figure 8: ECDF of 10,000 reducing attacks under different values of n and the corresponding attack success rates.

We conduct reducing attacks on three datasets, and the results show an average EER increase of 2.40% and a WER increase of 16.83% after the attack, as shown in Table 5. This indicates that anonymity improves and speech quality decreases, rendering the attack ineffective. These experiments demonstrate that the multi-parameter warping function exhibits strong robustness against attacks attempting to reverse or weaken the voice conversion effect by guessing parameters.

Comparison of voice conversion methods. We compare the multi-parameter warping function with three recent voice conversion methods: VoiceMask [11, 12], McAdams [17] and Overo [13]. We select these methods because they operate swiftly without user registration or pre-trained models and can be reversed when the parameters are known, similar to our approach. Additionally, they are commonly used as baselines

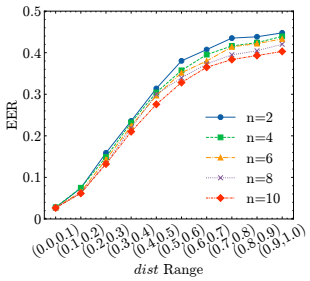


Figure 9: EER variation with *dist* range for different *n* values.

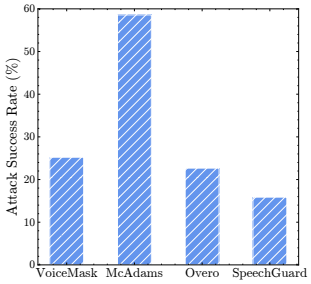


Figure 10: Comparison results of 10,000 reducing attacks.

Table 5: Results of reducing attacks on three datasets.

Dataset	EER (%)			WER (%)		
	Ori.	Pro.	Att.	Ori.	Pro.	Att.
LS test-clean	1.47	34.50	36.77	2.46	4.80	18.39
VCTK	1.46	35.66	37.22	6.58	14.81	32.69
Volunteers	2.21	30.29	33.66	6.35	10.97	29.99

Ori.: Original utterances, **Pro.**: Utterances after protection, **Att.**: Utterances after attack.

Table 6: RTC for acoustic privacy protection and recovery.

	VoiceMask	McAdams	Overo	SPEECHGUARD
Pro.	0.29	0.27	0.30	0.32
Rec.	0.34	0.21	0.31	0.26

Pro.: Operations to protect original utterances, **Rec.**: Operation to recover protected utterances to their original form.

for acoustic privacy protection [60, 61].

We simulate 10,000 reducing attacks on these four methods. Figure 10 shows the success rates of these attacks. The results indicate that SPEECHGUARD achieves the lowest attack success rate, which is 6.80% lower than the next best method, Overo, demonstrating the highest resistance to reducing attacks.

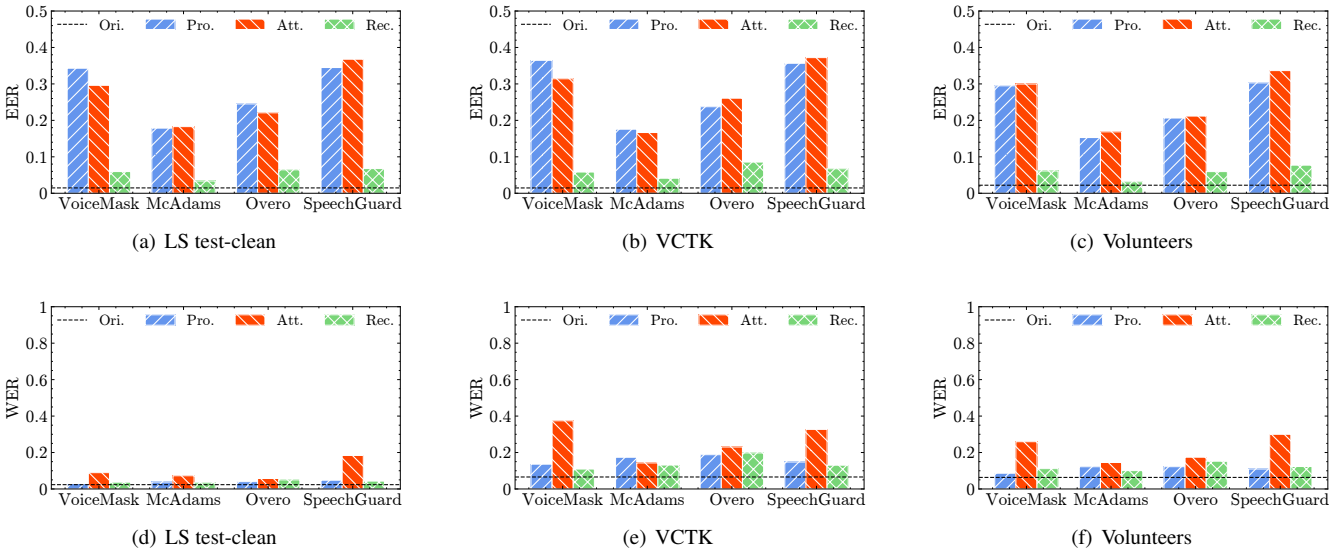
We use the four methods to convert utterances in three datasets. Figure 11 shows the EER and WER comparison results of the four methods after protection, attack, and recovery. Compared with other methods, the average EER of speech protected with SPEECHGUARD (33.48%) is comparable to that of Voicemask (33.42%) and significantly higher than those of McAdams (16.91%) and Overo (22.97%). SPEECHGUARD consistently achieves the highest EERs after reducing attacks (35.88%) compared to the protected version, whereas the other three methods exhibit cases where the EER is lower than the protected version in at least one dataset. The average WER of speech protected with SPEECHGUARD (10.19%) is comparable to other methods, and the WER increases significantly after the reducing attack (to 27.02%), making the attack ineffective due to the worse speech quality. Consequently, SPEECHGUARD can produce higher anonymity with similar speech quality. After reducing attacks, SPEECHGUARD produces the highest error rates in ASV and ASR, indicating the best robustness to such attacks. After recovery, all methods show good recovery in EER (SPEECHGUARD: 7.07%) and WER (SPEECHGUARD: 9.90%), but there is still a certain gap compared to the original speech. Although the protection is theoretically reversible, actual implementation operations such as interpolation result in differences from the original results.

Table 6 records the RTC for acoustic privacy protection and recovery using the four methods. The results show that the RTC for these methods is not much different. SPEECHGUARD takes a bit longer to protect acoustic privacy mainly because it requires generating more warping parameters within the appropriate range to enhance security.

5.3 Content Privacy

Components used to protect content privacy include the detection and encryption of sensitive texts. We evaluate the effectiveness of this protection by checking whether encrypted sensitive texts can still be recognized. We conduct experiments on speech in both WAV and MP3 formats. To evaluate the MP3 format, we convert WAV files to MP3 using a 24 Kbps VBR setting. Furthermore, we evaluate the time performance of this component as the number of sensitive words or patterns and the number of frames to encrypt vary. In this section, we focus solely on automatic sensitive text detection.

We randomly select 5 words from the utterances for each speaker to form a list of sensitive words. The total number of occurrences of these words in all utterances of a speaker is considered the ground truth. We encrypt the identified sensitive words and replace them with silent frames of the same duration. Then, we compare the FNR between the number of occurrences of sensitive words detected by ASR in the original utterances, encrypted utterances, and decrypted utterances across the three datasets in two file formats, as shown in Table 7. The original utterances exhibit very low FNR (1.05%) on the three datasets. This indicates that almost all sensitive words can be detected in ASR transcripts. After encrypting these words, the FNR increases significantly (to 98.70%). The higher FNR signifies that more predefined sensitive words become undetectable, reflecting superior protection performance. Once decrypted, the utterances are recovered identically to their original state, allowing the key holder to hear sensitive words that would be difficult for someone without the key to deduce. The FNR of encrypted MP3 files (98.70%) is slightly higher than that of encrypted WAV files (97.38%), indicating better confidentiality. This is because the rounding operation during sensitive frame calculation allows for greater tolerance of alignment inaccuracies. Additionally, the time required for encryption and decryption in the MP3 format (RTC: 0.001) is lower than in the WAV format (RTC: 0.003)



(a) LS test-clean (b) VCTK (c) Volunteers

(d) LS test-clean (e) VCTK (f) Volunteers

Figure 11: Comparison of EER and WER across four voice conversion methods after protection, attack, and recovery. **Ori.**: Original utterances, **Pro.**: Utterances after protection, **Att.**: Utterances after attack, **Rec.**: Utterances after recovery. SPEECHGUARD provides the highest anonymity with comparable speech quality and shows the best resistance to reducing attacks.

Table 7: Results of content privacy protection and recovery.

	Dataset	FNR (%)			RTC		
		Ori.	Enc.	Dec.	Det.	Enc.	Dec.
WAV	LS test-clean	0.62	95.88	0.62	0.53	0.005	0.005
	VCTK	0.81	99.50	0.81	0.28	0.001	0.001
	Volunteers	1.73	96.75	1.73	0.55	0.003	0.003
MP3	LS test-clean	0.62	97.95	0.62	0.53	0.002	0.002
	VCTK	0.81	99.96	0.81	0.28	0.001	0.001
	Volunteers	1.73	98.19	1.73	0.55	0.001	0.001

FNR: **Ori.**: Original utterances, **Enc.**: Utterances after encryption, **Dec.**: Utterances after decryption. **RTC**: **Det.**: Automatic sensitive text detection, **Enc.**: Encryption, **Dec.**: Decryption.

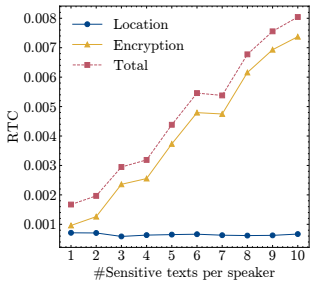


Figure 12: RTC changes with the number of sensitive texts per speaker.

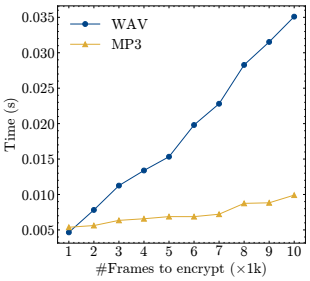


Figure 13: Encryption time changes with frame numbers in sensitive texts.

since fewer bytes are encrypted after encoding. After encryption, the size of the MP3 file is only 10% of the size of the WAV file, optimizing storage and transfer.

We measure the RTC of location and encryption as the number of sensitive words or patterns selected by each speaker changes, as shown in Figure 12. The RTC of location remains almost unchanged, while the RTC of encryption shows a near-linear increase as the number of sensitive texts per speaker grows. Compared to the transcription time of ASR, the time required for location and encryption is negligible. When the number of sensitive texts is small, encryption does not become a time bottleneck. The time required for encryption scales approximately linearly with the total number of frames in the texts to be encrypted, with a faster increase observed in WAV format compared to MP3, as shown in Figure 13. Note that encrypting phrases or sentences includes not only the frames of the words that make them up but also the inter-word gap frames, and the encryption time is the sum of these.

Comparison of sensitive text protection methods. We compare SPEECHGUARD's content privacy protection with three recent methods: Voicemask [11, 12], Preech [10], and Overo [13]. Voicemask employs an evolution-based algorithm to identify and replace sensitive words with words from the same category. Preech, Overo, and SPEECHGUARD use ASR, SD, or NER models for sensitive text detection. Preech discards them, Overo replaces them with a 1kHz sine wave, and SPEECHGUARD encrypts them. We randomly select a sensitive word for each speaker and apply SPEECHGUARD along with the three comparison methods to the three datasets. The comparison results are presented in Table 8. SPEECHGUARD has the same matching rate as Preech and Overo (98.13%),

Table 8: Comparison of content privacy protection and recovery across four methods.

Dataset	Voicemask				Preech				Overo				SpeechGuard			
	MR (%)	FNR (%)	RTC	REC	MR (%)	FNR (%)	RTC	REC	MR (%)	FNR (%)	RTC	REC	MR (%)	FNR (%)	RTC	REC
LS test-clean	83.82	83.82	0.59	✗	99.45	96.52	0.53	✗	99.45	97.13	0.53	✗	99.45	97.95	0.53	✓
VCTK	80.24	80.24	0.55	✗	98.42	96.44	0.28	✗	98.42	95.13	0.28	✗	98.42	99.96	0.28	✓
Volunteers	76.52	76.52	0.61	✗	96.51	95.02	0.55	✗	96.51	94.68	0.55	✗	96.51	95.59	0.55	✓

MR: Match Rate, **REC**: Recoverability.

Table 9: Results of user study.

Dialogue	Manual Sensitive Text Annotation								Mean Opinion Score						
	SD			Content Editing			Total RTC	Usability Rating	Protection			Recovery			Overall
	DER (%)	#User Corrections	RTC	#Auto	#Extra	RTC			Acoustic	Content	Speech Quality	Acoustic	Content	Speech Quality	
1	15.72	21±3	0.75	14	7±5	0.40	1.15	4.2	4.5	4.5	3.8	4.5	5.0	3.8	4.2
2	18.56	22±5	1.02	6	4±3	0.47	1.49	3.8	4.4	4.2	4.5	4.8	5.0	4.0	4.4
3	13.32	6±1	0.24	5	2±2	0.61	0.85	4.5	4.4	4.8	4.6	4.6	5.0	4.8	4.4

surpassing Voicemask (80.19%) because all three use the Conformer model for transcription, which is more accurate than direct speech signal comparison using dynamic time warping [62]. SpeechGuard consistently maintains the highest FNR (97.83%), indicating superior protection. The average RTC for SpeechGuard, Preech, and Overo (0.45) is lower than that of Voicemask (0.58), because the transcription model utilizes a GPU. However, this is strongly correlated with GPU performance and varies significantly across different datasets. Notably, while Preech, Overo, and SpeechGuard exhibit similar overall performance, only SpeechGuard allows recovery to the original version through decryption.

5.4 User Study

To evaluate the impact of manual annotation of sensitive texts on content protection and to assess user satisfaction with the protection and recovery performance of SpeechGuard, we conduct a user study.

Setup. In this study, we use three dialogues from after-sales service scenarios provided by NIO [63]. These scenarios include a 4S car inspection (5 minutes), roadside assistance (5 minutes), and after-sales maintenance consultation (2 minutes). Company insiders re-record these dialogues, replacing all users' personal information with fictitious details to ensure privacy. We input the dialogues into SpeechGuard for real-time voice conversion and automatic sensitive text detection. A user interface is then provided to 20 volunteers, who are tasked with reviewing the materials and determining necessary revisions, including newly edited texts and corrections to the model's output results.We record the time each volunteer takes to revise each audio clip. We combine automatic detection and manual annotation of sensitive texts for content privacy protection in speech and then recover the protected speech to its original state. Volunteers are asked to rate the usability of manual annotation, the system's protection and recovery performance, and overall performance on a five-point scale.[1]

Results. Table 9 summarizes the results. The initial average diarization error rate (DER) of the three dialogues is 15.87%. Users correct SD errors 21, 22, and 6 times, and add 7, 4, and 2 new edits respectively. The time required for users to correct errors correlates with the number of speaker changes within the conversation. The more re-labeling is needed, the longer it takes on average. We observe that while the initial automated results detect names, addresses, contact information, and license plate numbers mentioned in the conversation, users want further editing for some overlooked patterns and sentences. For example, a car price mentioned by a user at a dealership might reflect the user's income level, or a sentence like, "Yes, I am driving when I suddenly hear a strange engine noise, and then the car stalls." Users have their own perception of privacy, and the location and number of edits vary from person to person. Overall, users rate the manual annotation usability as generally "good", with most agreeing it should remain an option. They rate the protection, recovery, and overall performance of SpeechGuard as generally "good" or above, particularly praising the recovery performance, which aligns well with our original design goals.

5.5 Comprehensive Evaluation

We follow the steps outlined in Figure 2 for the entire SpeechGuard process, excluding manual annotation, and provide both WAV and MP3 formats. We randomly splice audio from different speakers in the LibriSpeech test-clean dataset, creating 3000 simulated dialogues. Each dialogue involves two speakers, designated as Spk_A and Spk_B. Additionally, two different sensitive texts are randomly selected for each conversation, referred to as $Text_A$ and $Text_B$. We establish four permission groups, labeled $Group_A$, $Group_B$, $Group_C$, and

[1] A 5-point scale (1 = bad; 2 = poor; 3 = fair; 4 = good; 5 = excellent).

Table 10: Results of comprehensive evaluation.

Permission		FNR (%)	EER (%)	WER (%)
Original		1.29	0.76	2.36
WAV	$Group_A$	$Text_A$: 1.32, $Text_B$: 99.67	Spk_A: 7.27, Spk_B: 35.32	5.04
	$Group_B$	$Text_A$: 98.09, $Text_B$: 0.68	Spk_A: 34.29, Spk_B: 7.69	4.88
	$Group_C$	1.29	7.78	4.32
	$Group_D$	98.18	34.50	5.30
MP3	$Group_A$	$Text_A$: 1.86, $Text_B$: 99.45	Spk_A: 7.33, Spk_B: 35.37	4.84
	$Group_B$	$Text_A$: 100, $Text_B$: 0.59	Spk_A: 36.75, Spk_B: 7.45	3.54
	$Group_C$	1.29	7.59	3.31
	$Group_D$	99.56	36.00	5.34

$Group_D$. $Group_A$ is given access to all warping parameters for conducting voice conversion on Spk_A and the encryption keys for $Text_A$. This allows them to recover the voice of Spk_A and decrypt $Text_A$, but they cannot revoke the protection imposed on Spk_B and $Text_B$. The permissions assigned to $Group_B$ complement those of $Group_A$. Both $Group_A$ and $Group_B$ belong to L_2 in the threat model. They can only recover sensitive information within their own permissions and cannot access each other's sensitive information. $Group_C$ is granted access to all warping parameters and encryption keys, enabling them to reverse both acoustic and content protection, and they belong to L_1. $Group_D$ does not have access to any sensitive information and can only listen to fully protected audio, belonging to L_3. Table 10 presents the experimental results, confirming the capabilities of the three permission levels. The MP3 format exhibits slightly higher protection performance than the WAV format while significantly reducing the required storage space. Compared to audio files, the space occupied by parameters and keys is negligible. Both the WAV and MP3 formats have a protection RTC of 0.86 and a recovery RTC of 0.27.

6 Related Work

Security & privacy risks in speech data publishing. Sensitive information contained in speech data can be exposed and misused. Attackers can easily train models on public speech datasets to perform de-anonymization attacks and infer individuals' identities [64–66]. Speech synthesis [14–16] enables attackers to convincingly imitate people's voices for identity theft. With only a few voice samples, attackers can deceive voiceprint authentication systems and gain access to users' sensitive information [14, 67, 68]. Speech content may also pose privacy risks. Attackers can use natural language processing techniques to analyze semantic details and infer the speaker's gender, age, race, origin, health, and social status [69, 70], and even reveal the speaker's emotions [71–73] and intentions [74, 75]. Therefore, we believe that both acoustic and content privacy should be protected before speech data is published.

Voice anonymization. Voice anonymization is a method to protect acoustic privacy, involving techniques such as linear pitch scaling [76], VTLN [18], and spectral mapping [77, 78]. Linear pitch scaling, which alters the pitch by shifting frequencies or stretching time, is often considered too simplistic. VTLN uses nonlinear distortion functions, such as the bilinear function [21] and the quadratic function [22], to distort the frequency axis. Our proposed multi-parameter warping function is also a VTLN-based approach. Spectral mapping modifies an individual's voice to mimic a specific target person. Unlike pitch scaling and VTLN, spectral mapping requires a pre-registration process to learn the target speaker's spectrum and cannot easily adapt to new speakers. Moreover, it cannot recover the source speaker's voice unless the inverse mapping is learned during the registration phase.

Speech content editing. The most direct way to protect speech content privacy is to remove or replace sensitive texts with nonsensical sounds. Some APIs and software can automatically identify and remove sensitive texts [79–81] by converting speech to text, searching for specific patterns, and returning edited speech. However, these tools are limited to predefined patterns, mostly personal identification information such as names, ID numbers, and addresses. Manual editing [82–84] offers more flexibility, enhancing user-specific privacy. Some methods synthesize similar words to replace sensitive ones [10, 11]. However, these techniques result in irreversible changes to the original speech content.

7 Conclusion & Future Work

In this study, we present the design, implementation, and evaluation of SpeechGuard, a real-time speech privacy-preserving system offering recoverable and customizable protection for acoustic and content privacy. Audio owners can adjust privacy definitions, protection strength, and listener permissions according to their needs and can recover the original version under specific permissions. The system supports lossy formats like MP3. We conduct extensive evaluations of the system in terms of security, privacy, speech quality, and computational overhead, using state-of-the-art ASV and ASR models on three English datasets, and we conduct a user study, confirming the system's practicality and applicability.

However, this study has some limitations that can be addressed in future work. First, the parameter range for voice conversion is determined empirically and validated only on the three aforementioned English datasets. Future research will expand this to other languages. Second, the accuracy of the offline SD and NER models used by the system significantly impacts the privacy-preserving effectiveness and user experience, making model accuracy a key focus for future improvements. Additionally, the system's automatic sensitive text detection currently supports only precise sensitive words and patterns. Future work could enhance detection capabilities for sensitive sentences and paragraphs by incorporating advanced NLP methods and contextual analysis.

Acknowledgments

We thank the anonymous reviewers for their valuable and constructive feedback. This research is partially supported by National Key R&D Program of China under Grant No. 2021ZD0110400, Innovation Program for Quantum Science and Technology 2021ZD0302900 and China National Natural Science Foundation with No. 62132018, 62231015, "Pioneer" and "Leading Goose" R&D Program of Zhejiang, 2023C01029 and 2023C01143, and NIO Inc.

References

[1] Chang Zeng, Xin Wang, et al. Attention back-end for automatic speaker verification with multiple enrollment utterances. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6717–6721, 2022.

[2] Xiaoyi Qin, Na Li, et al. Simple attention module based speaker verification with iterative noisy label detection. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6722–6726, 2022.

[3] Saeid Safavi, Martin Russell, et al. Automatic speaker, age-group and gender identification from children's speech. *Computer Speech & Language*, 50:141–156, 2018.

[4] Premjeet Singh, Md Sahidullah, et al. Modulation spectral features for speech emotion recognition using deep neural networks. *Speech Communication*, 146:53–69, 2023.

[5] Bloomberg. Amazon workers are listening to what you tell alexa, 2019.

[6] MUO. 4 big problems with alexa voice shopping and how to fix them, 2017.

[7] The Guardian. Hey Siri! Stop recording and sharing my private conversations, 2019.

[8] Film ratings. https://www.motionpictures.org/film-ratings/.

[9] Interpretation of the supreme people's court on the application of the civil procedure law of the people's republic of china. https://www.court.gov.cn/jianshe-xiangqing-353651.html.

[10] Shimaa Ahmed, Amrita Roy Chowdhury, et al. Preech: A system for Privacy-Preserving speech transcription. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2703–2720. USENIX Association, August 2020.

[11] Jianwei Qian, Haohua Du, et al. Speech sanitizer: Speech content desensitization and voice anonymization. *IEEE Transactions on Dependable and Secure Computing*, 18(6):2631–2642, 2021.

[12] Jianwei Qian, Haohua Du, et al. Hidebehind: Enjoy voice input with voiceprint unclonability and anonymity. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, SenSys '18, page 82–94, New York, NY, USA, 2018. Association for Computing Machinery.

[13] Jaemin Lim, Kiyeon Kim, Hyunwoo Yu, and Suk-Bok Lee. Overo: Sharing private audio recordings. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1933–1946, New York, NY, USA, 2022. Association for Computing Machinery.

[14] Ye Jia, Yu Zhang, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 4485–4495. Curran Associates Inc., 2018.

[15] Yuxuan Wang, R. J. Skerry-Ryan, et al. Tacotron: Towards end-to-end speech synthesis. In *Interspeech*, 2017.

[16] Nal Kalchbrenner, Erich Elsen, et al. Efficient neural audio synthesis. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2410–2419. PMLR, 10–15 Jul 2018.

[17] N. Tomashenko, Brij Mohan Lal Srivastava, et al. Introducing the VoicePrivacy Initiative. In *Proc. Interspeech 2020*, pages 1693–1697, 2020.

[18] D. Sundermann and H. Ney. Vtln-based voice conversion. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795)*, pages 556–559, 2003.

[19] E. Eide and H. Gish. A parametric approach to vocal tract length normalization. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 346–348 vol. 1, 1996.

[20] Maulik C. Madhavi and Hemant A. Patil. Vocal tract length normalization using a gaussian mixture model framework for query-by-example spoken term detection. *Computer Speech & Language*, 58:175–202, 2019.

[21] A. Acero and R.M. Stern. Robust speech recognition by normalization of the acoustic space. In *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, pages 893–896 vol.2, 1991.

[22] M. Pitz and H. Ney. Vocal tract normalization equals linear transformation in cepstral space. *IEEE Transactions on Speech and Audio Processing*, 13(5):930–944, 2005.

[23] John Bethencourt, Amit Sahai, et al. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334, 2007.

[24] Yang Yang, Jianguo Sun, et al. Practical revocable and multi-authority cp-abe scheme from rlwe for cloud computing. *Journal of Information Security and Applications*, 65:103108, 2022.

[25] Chandan Kumar Chaudhary, Richa Sarma, et al. Rmacpabe : A multi-authority cpabe scheme with reduced ciphertext size for iot devices. *Future Generation Computer Systems*, 138:226–242, 2023.

[26] Zhengyang Chen, Sanyuan Chen, et al. Large-scale self-supervised speech representation learning for automatic speaker verification. *CoRR*, abs/2110.05777, 2021.

[27] Shobha Bhatt, Anurag Jain, et al. Acoustic modeling in speech recognition: A systematic review. *International Journal of Advanced Computer Science and Applications*, 11, 2020.

[28] Isa Maks and Piek Vossen. A lexicon model for deep sentiment analysis and opinion mining applications. *Decision Support Systems*, 53(4):680–688, 2012.

[29] Ramya Rasipuram and Mathew Magimai-Doss. Acoustic and lexical resource constrained asr using language-independent acoustic model and language-dependent probabilistic lexical model. *Speech Communication*, 68:23–40, 2015.

[30] Yue Weng, Sai Sumanth Miryala, et al. Joint contextual modeling for asr correction and language understanding. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6349–6353, 2020.

[31] Chengyu Wang, Suyang Dai, et al. Arobert: An asr robust pre-trained language model for spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1207–1218, 2022.

[32] Xian Tang. Hybrid hidden markov model and artificial neural network for automatic speech recognition. In *2009 Pacific-Asia Conference on Circuits, Communications and Systems*, pages 682–685, 2009.

[33] George E. Dahl, Dong Yu, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[34] Geoffrey Hinton, Li Deng, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[35] Quan Wang, Carlton Downey, et al. Speaker diarization with lstm. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243, 2018.

[36] Hervé Bredin, Ruiqing Yin, et al. Pyannote.audio: Neural building blocks for speaker diarization. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7124–7128, 2020.

[37] Nithin Rao Koluguri, Taejin Park, et al. Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8102–8106, 2022.

[38] Jing Li, Aixin Sun, et al. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2022.

[39] Chen Liang, Yue Yu, et al. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1054–1064, New York, NY, USA, 2020. Association for Computing Machinery.

[40] Zoom. https://zoom.us/.

[41] Microsoft teams. https://www.microsoft.com/en-us/microsoft-teams/join-a-meeting.

[42] Tencent meeting. https://meeting.tencent.com/.

[43] Efrain Zenteno and Manuel Sotomayor. Robust voice activity detection algorithm using spectrum estimation and dynamic thresholding. In *2009 IEEE Latin-American Conference on Communications*, pages 1–5, 2009.

[44] Vassil Panayotov, Guoguo Chen, et al. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

[45] Mirco Ravanelli, Titouan Parcollet, et al. SpeechBrain: A general-purpose speech toolkit, 2021.

[46] Art. 9 gdpr - processing of special categories of personal data. https://gdpr-info.eu/art-9-gdpr/.

[47] Michael McAuliffe, Michaela Socolof, et al. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, 2017.

[48] ReadBeyond. Aeneas. https://github.com/readbeyond/aeneas, March 2017.

[49] Jacob Devlin, Ming-Wei Chang, et al. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[50] Alexei Baevski, Henry Zhou, et al. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477, 2020.

[51] Wei-Ning Hsu, Benjamin Bolte, et al. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *CoRR*, abs/2106.07447, 2021.

[52] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, and Furu Wei. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *CoRR*, abs/2110.13900, 2021.

[53] M. Dworkin, E. Barker, et al. Advanced encryption standard (aes). Federal Information Processing Standards (NIST FIPS) NIST FIPS 197, National Institute of Standards and Technology, Gaithersburg, MD, 2001.

[54] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92), 2019.

[55] Brecht Desplanques, Jenthe Thienpondt, et al. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. In *Interspeech 2020*, interspeech 2020. ISCA, October 2020.

[56] David Snyder, Daniel Garcia-Romero, et al. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018.

[57] Anmol Gulati, James Qin, et al. Conformer: Convolution-augmented transformer for speech recognition, 2020.

[58] A. Nagrani, J. S. Chung, et al. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[59] J. S. Chung, A. Nagrani, et al. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

[60] The voiceprivacy 2022 challenge. https://www.voiceprivacychallenge.org/.

[61] Jiangyi Deng, Fei Teng, et al. V-Cloak: Intelligibility-, naturalness- & Timbre-Preserving Real-Time voice anonymization. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5181–5198, Anaheim, CA, August 2023. USENIX Association.

[62] Guoguo Chen, Carolina Parada, and Georg Heigold. Small-footprint keyword spotting using deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4087–4091, 2014.

[63] Nio company. https://www.nio.cn/.

[64] Guangke Chen, Sen Chenb, et al. Who is real bob? adversarial attacks on speaker recognition systems. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 694–711, 2021.

[65] Domna Bilika, Nikoletta Michopoulou, et al. Hello me, meet the real me: Voice synthesis attacks on voice assistants. *Computers & Security*, 137:103617, 2024.

[66] Xuejing Yuan, Yuxuan Chen, et al. CommanderSong: A systematic approach for practical adversarial voice recognition. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 49–64, Baltimore, MD, August 2018. USENIX Association.

[67] Sercan Arik, Jitong Chen, et al. Neural voice cloning with a few samples. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[68] Jian Cong, Shan Yang, et al. Data Efficient Voice Cloning from Noisy Samples with Domain Adversarial Training. In *Proc. Interspeech 2020*, pages 811–815, 2020.

[69] Dan Gillick. Can conversational word usage be used to predict speaker demographics? In *Proc. Interspeech 2010*, pages 1381–1384, 2010.

[70] Nicholas Cummins, Alice Baird, et al. Speech analysis for health: Current state-of-the-art and the increasing impact of deep learning. *Methods*, 151:41–54, 2018. Health Informatics and Translational Data Analytics.

[71] Zheng Lian, Jianhua Tao, Bin Liu, Jian Huang, Zhanlei Yang, and Rongjun Li. Context-dependent domain adversarial neural network for multimodal emotion recognition. In *Interspeech*, 2020.

[72] Johannes Wagner, Andreas Triantafyllopoulos, et al. Dawn of the transformer era in speech emotion recognition: Closing the valence gap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10745–10759, 2023.

[73] Jeong-Yoon Kim and Seung-Ho Lee. Coordvit: A novel method of improve vision transformer-based speech emotion recognition using coordinate information concatenate. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4, 2023.

[74] Yue Gu, Xinyu Li, et al. Speech intention classification with multimodal deep learning. In Malek Mouhoub and Philippe Langlais, editors, *Advances in Artificial Intelligence*, pages 260–271, Cham, 2017. Springer International Publishing.

[75] Swayambhu Nath Ray, Minhua Wu, et al. Listen with intent: Improving speech recognition with audio-to-intent front-end. In *Interspeech 2021*. ISCA, aug 2021.

[76] Jean Laroche. *Time and Pitch Scale Modification of Audio Signals*, pages 279–309. Springer US, Boston, MA, 2002.

[77] Anisha Yathigiri, Meenalatha Bathula, et al. Voice transformation using pitch and spectral mapping. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1540–1544, 2017.

[78] Srinivas Desai, Alan W. Black, et al. Spectral mapping using artificial neural networks for voice conversion. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5):954–964, 2010.

[79] Text classification and redaction - google api. https://cloud.google.com/dlp/docs/concepts-text-redaction.

[80] Redacting or identifying personally identifiable information - amazon api. https://docs.aws.amazon.com/transcribe/latest/dg/pii-redaction.html.

[81] Vidizmo - ai with automatic redaction software. https://vidizmo.com/vidizmo-artificial-intelligence-solutions/redaction/.

[82] Adobe audition. a professional audio workstation. https://www.adobe.com/products/audition.html.

[83] Audacity. https://www.audacityteam.org/.

[84] Garageband. https://www.apple.com/mac/garageband/.