A privacy-preserving method using secret key for convolutional neural network-based speech classification

Shoko Niwa
Tokyo Metropolitan University
Tokyo, Japan
niwa-shoko@ed.tmu.ac.jp

Sayaka Shiota
Tokyo Metropolitan University
Tokyo, Japan
sayaka@tmu.ac.jp

Hitoshi Kiya
Tokyo Metropolitan University
Tokyo, Japan
kiya@tmu.ac.jp

***Abstract*—In this paper, we propose a privacy-preserving method with a secret key for convolutional neural network (CNN)-based speech classification tasks. Recently, many methods related to privacy preservation have been developed in image classification research fields. In contrast, in speech classification research fields, little research has considered these risks. To promote research on privacy preservation for speech classification, we provide an encryption method with a secret key in CNN-based speech classification systems. The encryption method is based on a random matrix with an invertible inverse. The encrypted speech data with a correct key can be accepted by a model with an encrypted kernel generated using an inverse matrix of a random matrix. Whereas the encrypted speech data is strongly distorted, the classification tasks can be correctly performed when a correct key is provided. Additionally, in this paper, we evaluate the difficulty of reconstructing the original information from the encrypted spectrograms and waveforms. In our experiments, the proposed encryption methods are performed in automatic speech recognition (ASR) and automatic speaker verification (ASV) tasks. The results show that the encrypted data can be used completely the same as the original data when a correct secret key is provided in the transformer-based ASR and x-vector-based ASV with self-supervised front-end systems. The robustness of the encrypted data against reconstruction attacks is also illustrated.**

***Index Terms*—Privacy preservation, Audio encryption, Automatic speech recognition, Automatic speaker verification**

I. Introduction

In recent years, cloud services have been increasingly used in many applications. Cloud services have the advantages of reducing initial computer investment and maintenance costs, and facilitating information sharing. However, since cloud services are managed by external providers, various threats such as data leakage due to malicious attacks from outside or inside are a concern [1]. When using classification models on a cloud service, it is necessary to provide a trained model and query data to the cloud service. Therefore, when cloud services are insecure, models and queries face threats. To prevent such risks, it is important to preserve privacy before sending data to insecure services.

Speech data usually includes personal information such as age, gender, language, and speaking content. Therefore, the issue of privacy also has been gradually gaining attention as the latest topic in the research field of speech processing [2]. In the research field of image processing, many privacy-preserving methods have been proposed for CNN-based systems [3], [4]. Some latest speech classification systems also adopt a convolutional layer for accepting speech data. Thus, the privacy-preserving methods for image classification can be easily applied to such CNN-based speech classification systems.

There are two patterns for inputting speech data into a convolutional layer: using a two-dimensional representation such as a spectrogram, and directly using a waveform. Therefore, we propose privacy-preserving methods that use a secret key to encrypt both spectrograms and waveforms and are assumed to have a random matrix with an invertible inverse. As examples of encryption methods with random matrices with invertible inverses, we propose two methods: *shuffling* and *flipping*. Speech data encrypted by either encryption method is highly distorted compared with its original data. Therefore, the classification task can only be performed correctly when a correct key is provided. In the experiments, we performed the proposed privacy-preserving methods in automatic speech recognition (ASR) and automatic speaker verification (ASV). From the results, we confirmed that when a correct secret key is used, the classification performances are completely the same as those without encryption, and when an incorrect key is used, the accuracies are significantly decreased. Additionally, in this study, we evaluate the difficulty of reconstructing the original information from the encrypted spectrograms and waveforms. Regarding sound reconstruction methods, phase reconstruction approaches and decryption attacks can be considered to reconstruct the original waveforms from spectrograms [5]. From these reconstruction experiments, the proposed methods can be shown to have high-security performance. To evaluate the difficulty of reconstructing the original spectrograms from the encrypted spectrograms and waveform, phase reconstruction and a decryption attack were performed on encrypted speech data.

In the following, we outline the structure of the paper. In Section II, we describe the privacy-preserving classification scenario. In Section III, we describe the details of the proposed

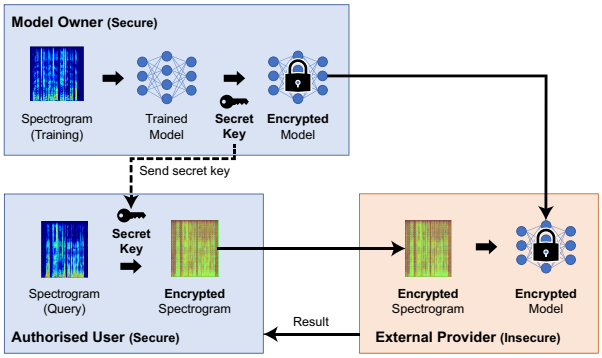
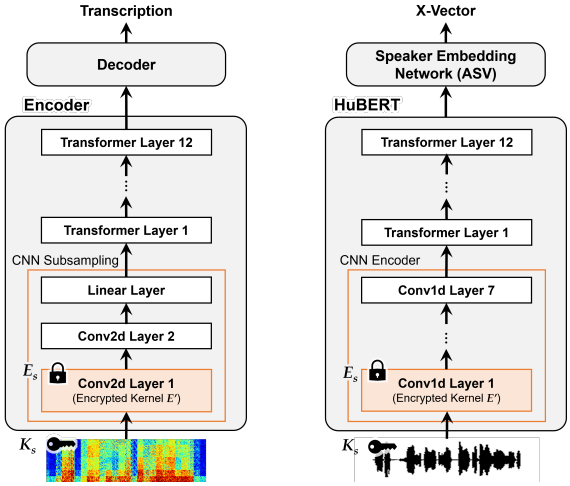


Fig. 1: Privacy-preserving scenario



(a) ASR system [9] (b) ASV system [10]

Fig. 2: Examples of models accepting encrypted speech data

method and the attack issue of the proposed method, and in Section IV we show the results. In Section V, we conclude the study and describe our future work.

II. Privacy-preserving classification scenario

Generally, there are three types of privacy-preserving issue for machine learning-based systems: (1) privacy of datasets, (2) privacy of models, and (3) privacy of models' outputs [6]. In this paper, we focus on speech classification considering privacy preservation in terms of the privacy of datasets. The privacy-preserving scenario is illustrated in Fig. 1. It is based on privacy preservation in image classification [7]. From Fig. 1, first, a model owner trains a classification model with plain speech data, e.g., spectrograms and waveforms in a secure environment. Then, the trained model is encrypted with a secret key. Since the encryption is performed after training, the secret key can be changed easily without retraining the model. Next, the model owner provides the encrypted model to an external provider, such as a cloud service, and shares the secret key with an authorized user. When the authorized user wants to use the encrypted classification system published by the external provider, an encrypted query with the shared secret key is sent. In this scenario, only an authorized user who knows the correct key can use the encrypted model. In comparison, an unauthorized user who does not know the correct key cannot use the model correctly and cannot reconstruct a speech utterance from the encrypted query. In this framework, it is assumed that the environment of model owners and authorized users is secure and that of external providers is insecure. Since an external provider performs classification by using the encrypted queries and models, the privacy information in spectrograms is protected even if the external provider is insecure.

III. Proposed methods

A. Query encryption

In this subsection, we describe a speech data encryption method that use a secret key. Basically, the procedure is followed to [8]. For adapting to speech data such as two-dimensional spectrograms, speech data is encrypted by whichever encryption method can be obtained through the following procedure. First, let speech data X be regarded as a spectrogram; speech data X with a size of $T \times F$ is divided into blocks with a size of $M \times M$ each, where T is the size of X in the time direction, F is the size of X in the frequency direction and M is the block size. Next, b-th block X_{b} is flattened into a one-dimensional vector with a size of M^2. Then, X_{b} is converted into X'_{b} by an encryption method while maintaining the dimension using a secret key. There are two methods for converting X_{b} to X'_{b} as follows;

1) Shuffling: A secret key K_{s} is an array of randomly permuted indices whose size is M^2, and its key space refers to the number of possible keys, denoted as $M^2!$. X_{b} is transformed into X'_{b} using K_{s} as follows:

$$X'_{\mathrm{b}}(i) = X_{\mathrm{b}}(K_{\mathrm{s}}(i)), \tag{1}$$

where $1 \leq i \leq M^2$. Eq. (1) shuffles the positions of the spectrogram values in X_{b} according to K_{s}. When the speech data X is regarded as a waveform, F_{b} is set to one in the procedure. In this case, the key space of K_{s} is $M!$, and the positions of the values included in the waveform can be shuffled by Eq. (1).

2) Flipping: A secret key K_{f} is a bit sequence, and zero and one are generated with equal probability. The size of K_{f} is M^2 and its key space is 2^{M^2}. X_{b} is transformed to X'_{b} using K_{f} as follows:

$$X'_{\mathrm{b}}(i) = \begin{cases} -X_{\mathrm{b}}(i) & (K_{\mathrm{f}}(i) = 1) \\ X_{\mathrm{b}}(i) & (K_{\mathrm{f}}(i) = 0) \end{cases}, \tag{2}$$

where $1 \leq i \leq M^2$. Eq. (2) inverts the sign of the spectrogram values in X_{b} according to K_{f}. When X is a waveform, the key space of K_{f} is 2^M and the transformation is performed according to Eq. (2).

Finally, each encrypted blocks X'_{b} is integrated to obtain the encrypted spectrogram X' with a size of $T \times F$.

B. Models accepting encrypted queries

To accept the encrypted queries with a secret key, the parameters of the trained classification model are transformed. The proposed encryption methods assume a classification model including a convolutional layer for accepting input data. When the kernel size and stride size of the convolutional layer are equal, it means that the input data is divided into patches without overlap. Since each encrypted block X'_{b} is treated independently, in this paper, the patch size is equal to the kernel and stride sizes. Let us denote the patch size of the convolutional layer to be encrypted as P, the number of output dimensions as d, and the block size as equal to the patch size; then, the kernel of the patch embedding layer can be expressed as $E \in \mathbb{R}^{P\times P\times d}$. To cancel out each encryption, the kernel of the model is defined as follows:

1) Shuffling: Kernel E is permuted with a shared secret key K_{s} so that it can correctly accept the encrypted data into a classification model. The encrypted kernel E' is defined using the permutation matrix E_{s}, which is defined with K_{s}, as follows:

$$E' = E_{\mathrm{s}}E. \tag{3}$$

We can treat encrypted data X' without retraining by simply attaching the encrypted kernel E' to the first convolutional layer.

2) Flipping: Inverting the sign of the weights contained in kernel E with a shared secret key K_{f} is necessary to correctly accept the encrypted data into the encrypted model. The transformation from kernel E into encrypted kernel E' is shown as follows:

$$E'[i,j] = \begin{cases} -E[i,j] & (k_l = 1) \\ E[i,j] & (k_l = 0) \end{cases}, \tag{4}$$

where $l = (i-1)P+j, 1 \leq i,j \leq P$ and k_l is the l-th element of K_{f}.

Figures 2(a) and 2(b) are examples of ASR and ASV models that were converted to accept data encrypted by shuffling. In this paper, we performed transformations on the first convolutional layer of each model. In the case of flipping, encrypted data X' can also be inputted without retraining by attaching the encrypted kernel E' in the same flow as in Fig. 2.

C. Attacks on encrypted data

To evaluate the security of encryption methods, it is necessary to demonstrate the difficulty of recovering the original data by performing decryption attacks. In the image processing research field, decryption attacks are performed on encrypted images to restore visual information on plain images [4]. In the speech processing research field, decryption attacks are performed on encrypted spectrograms or waveforms to restore the original data.

In this paper, to evaluate the privacy-preserving performance, we applied a phase reconstruction method and decryption attacks. The phase reconstruction method developed by Průša *et al.* [5], was adopted on the encrypted spectrograms to investigate whether the waveforms before encryption could be reconstructed. Since there are few attacks against encrypted spectrograms, we adopted Alex *et al.*'s [11] method of the decryption attack on encrypted images. The encrypted spectrograms are attacked as encrypted images.

TABLE I: WER(%) in the encryption scenario ($M = 3$) for ASR on LibriSpeech [12] (test_clean / test_other subsets).

	Plain	Correct key	Incorrect key
No encryption (Plain)	4.4 / 10.5	-	-
Shuffling	12.2 / 24.8	4.4 / 10.5	11.9 / 24.7
Flipping	97.8 / 98.4	4.4 / 10.5	97.8 / 98.1

TABLE II: EER(%) in the encryption scenario ($M = 10$) for ASV on VoxCeleb1 test set [13].

	Plain	Correct key	Incorrect key
No encryption (Plain)	8.3	-	-
Shuffling	41.3	8.3	37.6
Flipping	39.3	8.3	39.1

IV. Experiment

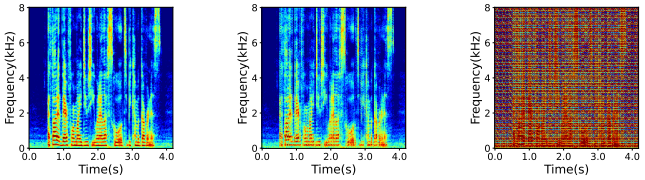
We evaluated the proposed privacy-preserving methods using ASR and ASV tasks.

A. Experimental conditions

For the ASR task, we trained a transformer model with the LibriSpeech corpus [12] following the ESPnet2 recipe [14]. The transformer architecture and hyperparameters were the same as in [9], except for the input feature and the stride size of the first convolutional layer. The input feature was set to 80-dim log-mel filterbank frames. The stride size of the first convolutional layer, which included the encrypted kernel E', was set to three in order to adopt the proposed method. The block size M for the encryption was set to three to match the kernel size of the first convolutional layer. Word error rate (WER) was used as an evaluation metric.

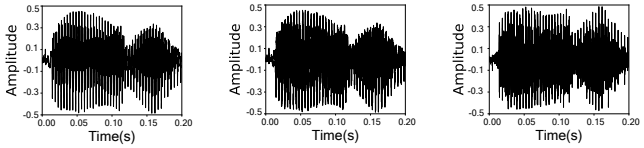
For the ASV task, we adopted an x-vector-based ASV system [15] with a self-supervised front-end model. A HuBERT model [10] trained with the LibriSpeech corpus was used as the self-supervised front-end model. The structure and hyperparameters of the HuBERT model were the same as those of the HuBERT BASE [10], except that the stride size of the first convolutional layer was changed to 10. The block size M for the encryption was set to 10. The speech expression outputted from the HuBERT model was inputted to the x-vector-based embedding network. The x-vector-based embedding network was trained with the VoxCeleb1 corpus [13], using the same hyperparameters as in [16]. Equal error rate (EER) was used as the evaluation metric.

We performed the proposed method in three scenarios: "Correct key", "Incorrect key", and "Plain". "Correct key" means that both keys used for encrypting the model and the queries were the same, "Incorrect key" means that both keys were not matched, and "Plain" means that only the model was encrypted, and the query was not encrypted.



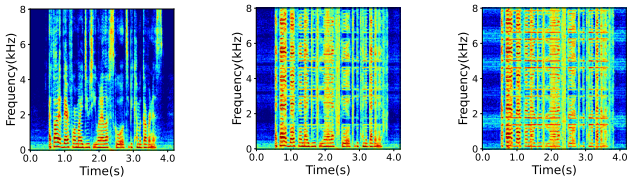
(a) Original (Plain) (b) Shuffling ($M = 3$) (c) Flipping ($M = 3$)

Fig. 3: Spectrograms encrypted with proposed method.



(a) Original (b) Shuffling ($M = 10$) (c) Flipping ($M = 10$)

Fig. 4: Waveforms encrypted by proposed method.



(a) Original (b) Shuffling ($M = 10$) (c) Flipping ($M = 10$)

Fig. 5: Spectrograms corresponding to Fig. 4.

B. Experimental results

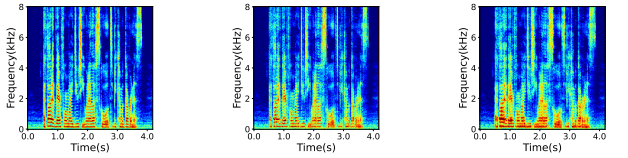
1) ASR task: Table I shows the results of the ASR experiments using the no-encryption (plain) model and the models encrypted by shuffling and flipping. The WERs of the encrypted models in Correct key were completely the same as those of the no-encryption model. This indicates that the Correct key scenario was performed as we expected. In the Incorrect key scenario, the WERs were higher than those of the Correct key scenario. In particular, when flipping was applied as the encryption method, the WERs increased significantly. To analyze these results, the spectrograms encrypted by the proposed method are shown in Fig. 3. Figure 3(a) shows the original plain spectrogram, and Figs. 3(b) and 3(c) show the spectrograms encrypted by shuffling and flipping, respectively, under the condition $M = 3$. By comparing Figs. 3(a) and 3(b), we can see that the positions of values in each block move in accordance with the secret key, and the harmonic structure of the spectrogram is distorted. By comparing Figs. 3(a) and 3(c), we can confirm that the magnitude of each value in the encrypted spectrogram changes randomly. The value in the spectrogram greatly increases owing to the sign inversion, especially in the silence intervals in Fig. 3(a), so the encrypted spectrogram is markedly different from the original one. A larger block size M results in a larger key space and better privacy-preserving performance. On the other hand, it will affect the accuracy of ASR in the Plain scenario, so there is a trade-off relationship between accuracy and privacy-preserving performance.

2) ASV task: Table II shows the results of the ASV experiments using the no-encryption (plain) model and the models encrypted by shuffling and flipping. Similarly to the ASR results, the EERs of the Correct key were completely the same as those of the no-encryption model. In the Incorrect key scenario, the EERs were higher than those of the the Correct key scenario. To analyze these results, the waveforms encrypted by the proposed method are shown in Fig. 4. Figure 4(a) shows the original plain waveform, Figs. 4(b) and 4(c) show the waveforms encrypted by shuffling and flipping, respectively, under the condition $M = 10$. Figures 5(a)-5(c) correspond to the spectrograms in Figs. 4(a)-4(c), respectively. By comparing the original and encrypted waveforms, we can see that there are changes in the outline, but they are minor. By contrast, from Figs. 5(b) and 5(c), it can be seen that frequency the response of the original waveform has been significantly changed by the encryption. These characteristics led to performance degradation without the correct key.

C. Evaluation of privacy-preserving performance

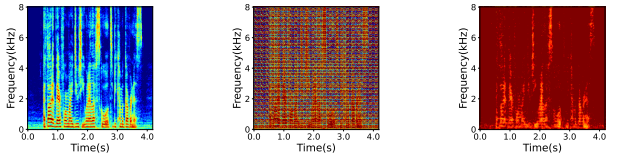
To evaluate the security of encryption methods, phase reconstruction was performed using Průša *et al.*'s method. Figure 6 shows the results of the phase reconstruction of the spectrogram encrypted by shuffling. We first applied shuffling to the spectrogram of the original speech (Fig. 6(a)) under $M = 3$. Then, we reconstructed the speech by applying phase reconstruction to the encrypted spectrogram (Fig. 6(b)). The spectrogram of the reconstructed speech is shown in Fig. 6(c). Figures 6(a) and 6(c) show that the structure of the spectrogram of the original speech and that of the speech obtained by phase reconstruction were different. The original speech and the speech obtained by phase reconstruction were also different. Figure 7 shows the results of the phase reconstruction of the spectrogram encrypted by flipping. Spectrograms in Fig. 7 were obtained by the same procedure as those for shuffling. Figures 7(a) and 7(c) show that the overall structure and amplitude values of the two spectrograms are significantly different. The original speech was hardly audible from the speech obtained by phase reconstruction. From the results in Figs. 6 and 7, we found that it is difficult to reconstruct the original speech from the spectrogram encrypted by the proposed methods.

Then, a decryption attack was performed on the encrypted spectrograms using Alex *et al.*'s method. The block size must be known when the attacker decrypts the encrypted data. Therefore, in the experiments, we assumed that the attacker knows the block size. Generally, spectrograms can be treated as grayscale images, but since Alex *et al.*' s method only supports 8-bit RGB images, we scale the spectrogram so that the maximum value is 255 and the minimum value is 0. Figure 8 shows the result of attacking a spectrogram encrypted by shuffling under the condition $M = 3$, and Fig. 9 shows the result of attacking a spectrogram encrypted by flipping under



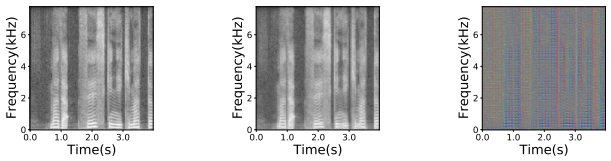
(a) Original (b) Encryption (c) Reconstruction

Fig. 6: Examples of phase reconstruction of the spectrogram encrypted by shuffling ($M = 3$)



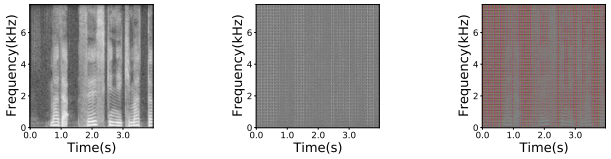
(a) Original (b) Encryption (c) Reconstruction

Fig. 7: Examples of phase reconstruction of the spectrogram encrypted by flipping ($M = 3$)



(a) Original (b) Encrypted (c) Decrypted

Fig. 8: Examples of decryption of the spectrogram encrypted by shuffling ($M = 3$)



(a) Original (b) Encrypted (c) Decrypted

Fig. 9: Examples of decryption of the spectrogram encrypted by flipping ($M = 3$)

the condition $M = 3$. From Figs. 8(c) and 9(c), we found that spectrograms could not be completely decrypted even if the attacker attacked with the existing method. Note also that this approach requires knowing the block size in advance. In this experiment, M was set to a small value to match the existing model structure. However, a larger M makes the proposed method even more robust because the key space is larger and decryption becomes more difficult.

V. Conclusion

In this paper, we proposed the privacy-preserving methods using a secret key for CNN-based models: shuffling and flipping. The encrypted spectrograms and waveforms obtained by the proposed methods were difficult to use without the correct key in the ASR and ASV tasks. In addition, to evaluate the privacy-preserving performance of the proposed encryption method, the phase reconstruction and decryption attack methods were applied to the encrypted data. Our experiments showed that only the authorized user who knows the correct key could use the classification system correctly. The robustness of the proposed methods against existing attack methods was also confirmed. As future work, we will develop a novel spectrogram and waveform encryption method that is less sensitive to block size and also further evaluate the robustness of the proposed method against existing decryption methods.

VI. Acknowledgements

This work was supported in part by SECOM Science and Technology Foundation.

References

[1] H. Tabrizchi and M. Kuchaki Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," *The journal of supercomputing*, vol. 76, no. 12, pp. 9493–9532, 2020.

[2] N. Tomashenko *et al.*, "The voiceprivacy 2022 challenge evaluation plan," [Online]. Available: https://www.voiceprivacychallenge.org/vp2020/docs/VoicePrivacy_2020_Eval_Plan_v1_4.pdf, 2020.

[3] H. Kiya, A. MaungMaung, Y. Kinoshita, S. Imaizumi, and S. Shiota, "An overview of compressible and learnable image transformation with secret key and its applications," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[4] A. Maungmaung and H. Kiya, "Privacy-preserving image classification using an isotropic network," *IEEE MultiMedia*, vol. 29, no. 2, pp. 23–33, 2022.

[5] Z. Průša, P. Balazs, and P. L. Søndergaard, "A noniterative method for reconstruction of phase from stft magnitude," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1154–1164, 2017.

[6] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310–1321, 2015. [Online]. Available: https://doi.org/10.1145/2810103.2813687

[7] H. Kiya, R. Iijima, A. Maungmaung, and Y. Kinoshita, "Image and model transformation with secret key for vision transformer," *IEICE Transactions on Information and Systems*, vol. E106.D, no. 1, pp. 2–11, 2023.

[8] R. Iijima and H. Kiya, "An encryption method of convmixer models without performance degradation," in *2022 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 159–164, 2022.

[9] S. Karita *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 449–456, 2019.

[10] W.-N. Hsu *et al.*, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[11] A. H. Chang and B. M. Case, "Attacks on image encryption schemes for privacy-preserving deep neural networks," 2020. [Online]. Available: https://arxiv.org/abs/2004.13263

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5206–5210, 2015.

[13] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230819302712

[14] S. Watanabe *et al.*, "Espnet: End-to-end speech processing toolkit," in *Proc. Interspeech 2018*, pp. 2207–2211, 2018. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[15] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

[16] S. wen Yang *et al.*, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, pp. 1194–1198, 2021.