LINGUARD: AUTHENTICATING SPEECH RECORDINGS USING SPEECH RECOGNITION AND WATERMARK

Shameer Faziludeen[1], *Arun Sankar M. S.*[2], *Phillip L. De Leon*[3], and *Utz Roedig*[1]

[1]University College Cork
School of Computer Science and Information Technology, Cork, Ireland
[2]South East Technological University
Department of Electronics and Communication Engineering, Carlow, Ireland
[3]University of Colorado Denver
Department of Electrical Engineering, Denver, Colorado, U.S.A.
{sfaziludeen,u.roedig}@ucc.ie, arun.sankar@setu.ie,
and Phillip.DeLeon@ucdenver.edu

ABSTRACT

We present LinGuard, a novel framework for protecting linguistic content of a speech recording using a signature of the speech transcription linked to the recording via an embedded watermark. The recording can be subject to modifications during normal processing (additive noise, coding, filtering) and the signature verification will not fail as long as these modifications do not alter the linguistic content. The signature verification fails if the linguistic content is changed by deletion, insertion, and substitution of a speech segment. Thus, tampering with recordings can be detected. Our LinGuard implementation makes use of AudioSeal (Watermarking), Falcon 512 (Signature), OpenAI Whisper (Speech Recognition) and models from SpeechBrain (Speaker Verification). We demonstrate the feasibility of this selection and evaluate the performance of these components in the context of LinGuard.

Index Terms— Watermarking, Automatic Speech Recognition (ASR), Speaker Verification (SV), Signature, Authentication

1. INTRODUCTION

The history of tampering with speech recordings is intertwined with the evolution of recording technology and audio processing tools. Our use of the word "tampering" is any technique which intentionally alters the linguistic content of the recording. There are at least three ways of tampering with a speech recording: deletion, insertion, and substitution of a segment. For example, insertion of a segment like "not" may completely change the meaning and context of the speech recording, e.g. "He did do that" vs. "He did not do that".

Detection of tampering in speech recordings has been studied extensively [1]. Signal analysis including identifying discontinuities in the waveform or spectrum [2], electrical network frequency (ENF) analysis [3], identifying changes in the microphone [4,5], and inconsistencies in the acoustic environment [6] can all be used to detect tampering. More recently, advances in AI-generated speech have led to widely-available and powerful tools for tampering such as voice clones which can generate entire sentences [7]. Thus, traditional tampering detection techniques become less effective. Although methods to detect AI-generated speech have been proposed and effectively used, rapid improvements in the quality of AI-generated speech continue to pose detection challenges [8].

Methods have been devised to make audio signals tamper-evident by design and may be classified as out-of-band and in-band solutions. Out-of-band solutions rely on additional data (metadata) to protect the speech signal. For example, a digital signature is distributed together with the recording and a listener can verify the signature and detect tampering. Such out-of-band methods have limitations namely that metadata may be lost when re-recording speech. In-band solutions such as watermarks, embed additional information into the audio signal which should be resilient to re-recording and edits. Our use of the word "edit" is any modification or processing, e.g. additive noise, coding, filtering, etc. which alters the signal but not necessarily the linguistic content of the recording. As an example, a signal segment which is missing the watermark would indicate tampering (substitution or insertion). However, we have recently demonstrated that substitution can be achieved by using watermarked speech segments [9].

Attempts have been made to combine cryptographic methods with watermarking. For example, Steinebach et al. [10] extract audio features such as root mean square (RMS) and Zero-crossing rate (ZCR). The features are then encrypted and embedded as watermark in the audio signal. This approach is resilient to some edits without causing a failure in the decryption. However, this approach offers limited protection as tampering with similar audio features is possible. At this time, existing methods do not link the linguistic content with the cryptographic material and the watermark.

There have also been recent approaches combining out-of-band and in-band solutions with cryptographic methods. The C2PA "Content Credentials" standard allows to cryptographically sign metadata and link it to an audio recording using a watermark [11]. If the signed metadata is stripped, the watermark lets a verifier recover it. However, this approach does not ensure the integrity of the linguistic content; the metadata is cryptographically secured but not the linguistic content.

In this work we present LinGuard, a novel approach of protecting speech recordings against tampering. In LinGuard we combine watermarking, digital signatures, ASR, and SV into a single system. The approach has the following properties:

(P1) It can be verified that the linguistic content is unaltered.

(P2) The speaker's voice in the recording can be verified.

(P3) Resilience to edits that do not modify the linguistic content.

In LinGuard, ASR is used to determine the text, i.e. linguistic content of the recording. A digital signature is created for the recording based on the text and linked to the recording using a robust watermark, satisfying (P3). A verifier can extract the watermark and

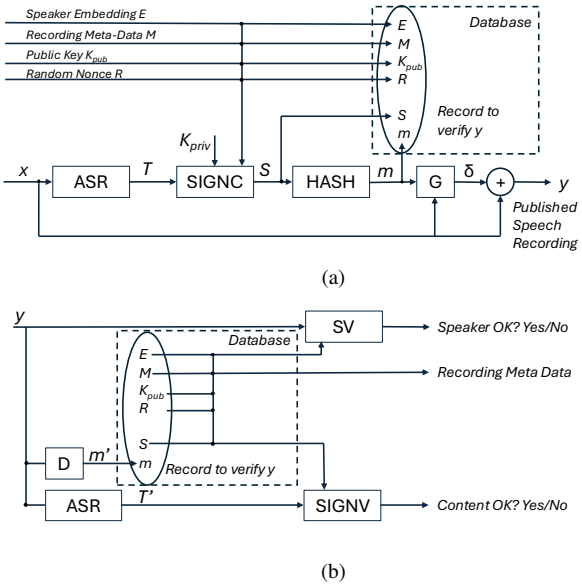

Fig. 1. LinGuard architecture: (a) signing a speech recording with steps S1–S6 (see Section 2.1) and (b) verifying linguistic content with steps V1–V4.

recover the signature and also use ASR to obtain the text. Now the verifier can check that the linguistic content matches the signature, satisfying (P1). Optionally, the verifier can use SV to ascertain that the voice matches the original speaker, satisfying (P2).

Implementing LinGuard is not without challenges. A watermark can only carry a limited amount of information, often a few bits, and must be present for a sufficient duration to be clearly detectable and robust. Furthermore, embedded watermarks should not compromise ASR or SV accuracy. In this work, we provide overall system design, outline watermarking procedures for embedding and recovering information necessary for verification, and evaluate tradeoffs.

2. LinGuard ARCHITECTURE AND IMPLEMENTATION

2.1. Architecture

Fig. 1 illustrates the overall architecture and the process used for signing and subsequently verifying a speech recording.

Signing Steps: We refer the reader to Fig 1a. (S1) A speaker speaks a sentence, the audio signal is denoted as x and the length is denoted L; (S2) ASR is used to transcribe the sentence into text T (the linguistic content); (S3) A digital signature S is created using K_{priv}. Input to the signature algorithm is T together with speaker embedding E, public key K_{pub}, random nonce R, and the optional meta-data M; (S4) A hash m of the signature S is created; (S5) The signature S together with all information used in the signature generation is stored out-of-band using hash m as index; (S6) The hash m is embedded as watermark δ in the speech recording which is released as y.

Verifying Steps: We refer the reader to Fig 1b. (V1) The hash m embedded as watermark in y is recovered; (V2) ASR is used to transcribe the speech into text T'; (V3) Using the text T' the recovered signature S and elements E, R, M and public key K_{pub} via m is verified using K_{pub}; (V4) A voice embedding E of the speaker and SV is used to verify the speaker identity.

The out-of-band storage may be implemented as blockchain or simple database. Signatory and verifiers must have access. Besides the signature S each record holds nonce R, the public key K_{pub} (or pointer to the key), a speech embedding E for SV(or a pointer to it) and additional metadata M relevant to the recorded speech (e.g. recording location, recording time, context of the recording). Each data record indexed by the hash m is protected against tampering via the signature. In case of using a blockchain as data store it is additionally ensured that entries cannot be altered once recorded.

The last verification step (V4) is optional and only necessary if (P2) is required. It may be omitted in cases where it is only relevant to ascertain that a speaker spoke these words but it is not necessary if the voice in the recording truly belongs to the speaker. For example, an adversary may record himself speaking the exact same phrase as the original speaker and then adds the watermark of the original recording. As ASR produces the same text transcription T' (V3) will accept the signature. SV requires a user to provide speaker embeddings which some users may be reluctant to do.

LinGuard protects the linguistic content robustly. The audio recording can undergo modification or processing (i.e. edits) such as additive noise, coding, filtering, etc. which alters the signal but not necessarily the linguistic content of the recording. Watermarking, ASR and SV are techniques proven to be robust.

2.2. Watermarking

To link a cryptographic signature inseparable with the voice recording we use a watermark. Numerous audio watermarking systems have been proposed [12–14] all with the goals of accurate detection and resilience to edits. In addition, watermarks may also carry a message and thus an additional goal is message recovery. Currently, AudioSeal [12] is the most advanced watermarking technique and it has been shown to outperform all other watermarking techniques [15]. Therefore we use AudioSeal in this work. We denote the watermark as:

$$\delta = G(x, m) \tag{1}$$

where x is the speech signal, m is a b-bit message, and $G(\cdot)$ is the watermark generator. The watermarked signal is thus given by:

$$y = x + \delta \tag{2}$$

the detector D outputs a tensor of shape $\mathbb{R}^{L,1+b}$:

$$\mathbf{p} = D\,(y) \tag{3}$$

where L is the duration of y (and x), b is the message length, and one bit for detection decision. A hard decision on the value of m and whether y is watermarked is obtained by time-averaging $\mathbf{p}$ and thresholding. For AudioSeal, $b = 16$.

2.3. Digital Signature

A digital signature is a cryptographic method for proving that data was created by a specific sender and has not been altered since it was signed [16]. The sender uses a private key K_{priv} to create the signature while a verifier use the corresponding public key K_{pub}. To avoid two identical inputs producing the same signature, a nonce R (a random number) can be used as part of the input.

Public key encryption algorithms such as RSA are currently being phased out due to vulnerability and being replaced with quantum-safe cryptography [17]. Available quantum-safe algorithms differ in terms of public key and signature size and also in computational complexity in terms of signing and verification. Both

classical and quantum algorithms require large signature sizes. For example, 2048b RSA signature size is 256B while Falcon 512 has a signature size of 666B. Some post quantum algorithms such as UOV Is-pkc has a short signature of 96B but requires a massive public key size of 66576B which is often considered impractical.

In the case of well-balanced algorithms such as Falcon 512, signatures are too large to directly embed these as an AudioSeal watermark within a speech signal. To deal with this limitation, we use a combined out-of-band and in-band approach. After generating the signature S we create a hash (a fingerprint) m of the signature using a hashing algorithm. This hash is used as the watermark message. The signature S is stored out-of-band e.g. within a blockchain or a database and the hash is used as an index to access the database record which holds the correct signature. We address the issue of the hash being longer than b bits through a segmentation approach described in the following subsection.

SHA-256 might be used to create a 32B hash recognizing that shorter signatures increase the collision probability. However, even if several candidate signatures are found only the correct one can pass verification. In this work we consider SHA-256 as example hashing algorithm. We propose Falcon 512 as signature algorithm but the algorithm choice has no impact on the evaluation results presented in this paper.

2.4. Segmentation

The hash m needs to be embedded within the speech signal y as watermark. A watermark is usually smaller than the hash embedded, for example, 16bit in case of AudioSeal vs. 256bit in case of SHA256. Thus, the hash m has to be divided into I segments m_i of duration D_S which have to be embedded as a sequence of watermarks ($I = 16$ in case of AudioSeal and SHA256). The segment length is variable and depends on the duration L of the speech that is protected by LinGuard.

Each watermark segment must be present for a sufficient duration L_S^{min} to reliably extract m_i. Thus, there is a minimum duration L^{min} for the spoken text T that can be protected by LinGuard. The minimum required duration of spoken text is calculated as:

$$L^{min} = I \cdot L_S^{min} \tag{4}$$

The 16 segments carrying the hash of the signature are spread over the entire duration L of the speech protected by the signature. The duration of the segment is proportional to the BER of m_i; longer segments reduce the error probability. It has to be noted that it is also possible to increase the number of segments I to include an error coding scheme (e.g. Fountain Code) which would increase L^{min}. However, we do not explore error coding in this work and address robustness by ensuring sufficient duration L_i^{min}. We determine via experimentation L^{min} in Section 3.1. The results show that $L_S^{min} > 300ms$ provide good results (a BER of at least 10^{-3}). This in turn means that the minimum length L^{min} must be at least 4.8s for LinGuard, allowing protection of speech recordings with meaningful linguistic content.

2.5. Automatic Speech Recognition (ASR)

ASR is used to determine the linguistic content embedded in the speech signal that is protected by LinGuard. In principle, any robust ASR can be used as a building block within LinGuard as long as ASR performance is not impacted by the presence of a watermark.

To ensure that each ASR processing step returns the same content we apply the following ASR post-processing steps: (SR1) convert all text to lower case, (SR2) remove any punctuation, (SR3) normalize whitespaces (convert multiple spaces to single), and (SR4) remove leading or trailing whitespace in order to produce T used to generate the signature S.

For the ASR component, we use the OpenAI Whisper toolbox which is a transformer-based, encoder-decoder model trained on more than 600k hours of data [18]. As we will show in Section 3.2, speech recognition accuracy is not impacted by the presence of an AudioSeal watermark.

2.6. Speaker Verification (SV)

An SV system is employed to confirm the ownership of a speech recording. From the large pool of SV systems available, ranging from traditional feature-based methods to advanced deep learning models, any of these can be adopted for the proposed framework. The only requirement is that the chosen SV system must demonstrate reliable performance on published speech recordings, which differ from raw speech signals due to the presence of watermarks.

We use the pre-trained models from SpeechBrain as these models are trained on large, diverse datasets and they can provide robust and generalizable representations or speaker embeddings using minimal training and limited data. For the SV component, we use SV models based on X-vector [19], ResNet [20], and ECAPA-TDNN [21]. As we will show in Section 3.3, SV accuracy is not impacted by the presence of an AudioSeal watermark. Any of these SV can be used within LinGuard.

3. EVALUATION

In this section we describe our evaluation consisting of: 1) the minimum segment duration L_S^{min} necessary to achieve reliable message recovery when using AudioSeal, 2) ASR accuracy in the presence of a watermark, and 3) SV accuracy in the presence of a watermark.

3.1. Minimum Segment Duration

To analyze the effect of segment duration on watermark detection and BER and determine L_S^{min}, we use a portion of the TIMIT [22] Corpus as follows. We randomly choose 160 speakers of the TIMIT test set each with 10 utterances, yielding a total of 1600 speech signals. If necessary, the speech signal is repeated so that the signal duration is at least 5s. We segment each speech signal into segments with fixed duration L_S; we consider segment durations between 1ms and 1000ms. We then randomly choose 100 segments and generate a watermark for each segment using a random message as in (1). The generated watermark is then added to the respective segment to create the watermarked speech segment as in (2) which is then passed to the watermark detector. Using the detector, we recover the message and compute BER. The resulting plot is shown in Figure 2, where we find that for segments longer than 200ms, we can achieve at least 10^{-3} BER for clean speech.

We repeated the experiment with additive noise in addition to the watermark so that the SNR = 35dB (PESQ = 3.5, where Perceptual Evaluation of Speech Quality (PESQ) is an objective alternative to subjective listening tests). For speech with noise added at 35dB SNR, the segment length needs to be more than 300ms to achieve at least 10^{-3} BER. We do not show the case with added noise in Figure 2 as lines would overlap; added noise has minimal impact on BER and is hardly perceptible.

For this work, $L^{\min} = 300$ms is a good choice allowing for some signal distortion while achieving a good BER.

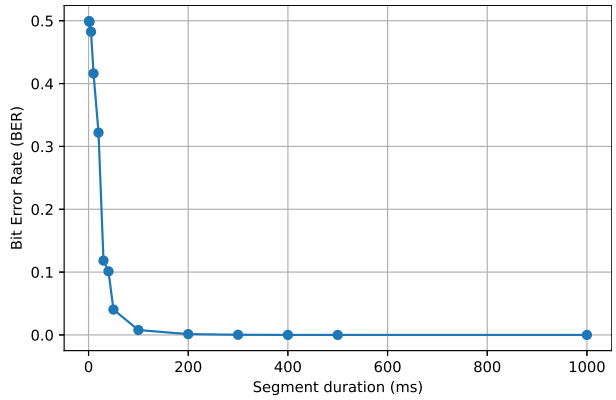

Fig. 2. Bit Error Rate (BER) for recovered watermark message with no noise. With segments longer than 200ms and 300ms for clean and noise added speech respectively, BER is less than or equivalent to 10^{-3}.

3.2. ASR Accuracy in the Presence of Watermark and Noise

We use OpenAI Whisper (default settings with the turbo model) for evaluation of ASR accuracy in the presence of watermark and noise. In our experiments, we use all speech recordings from 462 speakers from the TIMIT training set, with each speaker having 10 utterances each and measure the Word Error Rates (WER) and Character Error Rate (CER) by comparing TIMIT transcripts with Whisper ASR output. Processing steps as mentioned in Section 2.5 are performed prior to the performance evaluation. For each speech recording we use: (1) raw speech signals, (2) watermarked speech signals, and (3) watermarked speech signals with additive noise (SNR = 35dB). Table 1 shows WER and CER % results for the TIMIT speech recordings.

Table 1. ASR performance for raw, watermarked speech and watermarked speech with noise added.

	Raw speech	Watermarked speech	Watermarked speech with noise
WER (%)	2.89	2.92	2.93
CER (%)	0.91	0.91	0.92

As can be seen from the results the addition of an AudioSeal watermark does not impact significantly on performance. Similarly, additional noise also does not impact ASR significantly.

3.3. SV Accuracy in the Presence of Watermark and Noise

In order to evaluate SV accuracy in the presence of watermark and noise, we use three different systems: X-vector, ECAPA-TDNN, and ResNet. Each system is trained using 462 speakers from TIMIT training set to generate embeddings for each speaker. The SV is performed by comparing the embeddings of the test sample with those of the target speaker. The TIMIT material is used in the ratio of 80% for training and 20% for evaluation.

For evaluation we consider: (1) raw speech signals, (2) watermarked speech signals, and (3) watermarked speech signals with additive noise (SNR = 35dB). The results are provided in Table 2 where we find watermark and noise have negligible (if any) effect on SV accuracy. ResNet performs slightly better than the other evaluated solutions.

Table 2. Average SV accuracy in the presence of watermark and noise for three different systems.

Model	Accuracy (%)		
	Raw speech	Watermarked speech	Watermarked speech with noise
X-vector	98.34	98.48	98.05
ECAPA-TDNN	100	99.86	99.93
ResNet	100	99.86	100

4. DISCUSSION

We consider what happens if a signed recording is tampered with, by considering a deletion, insertion, and substitution of a portion of the signed speech recording. Because tampering intentionally alters the linguistic content of the recording, the tampered portion likely spans at least the duration of a phoneme which averages about 100ms.

If a phoneme or word is deleted from the recording and assuming the hash m can be recovered, then the text in the recording will not match the original text, i.e. $T^{'} \neq T$ and the linguistic content cannot be verified; if the duration of the deleted portion is large (similar to the segment size), the hash m may not be recovered due to increased bit errors (see Fig. 2) in which case the signature cannot be recovered and verification is not possible. Shorter deletions are unlikely to change the linguistic content thus signature recovery is possible and both recording and speaker can be verified.

We assume an attacker has access to a voice clone of the speaker and the watermark generator/detector in order to insert a phoneme or word with the same message. An attacker may also use dynamic time warping so signal insertion does not change the segment size compromising the segmentation procedure and recovery of the hash. Assuming the hash can be recovered, then the text in the recording will not match the original text, i.e. $T^{'} \neq T$ and the linguistic content cannot be verified. Without watermark generator access, the hash may not be recovered thus signature recovery is not possible; even if it is, the linguistic content cannot be verified. Shorter insertions are unlikely to change linguistic content thus signature recovery is possible and the recording and speaker can both be verified.

Finally, we consider a substitution equivalent to a deletion followed by insertion. Thus remarks for insertion apply to this case. There is also the possibility a word is replaced with the same word but different speaker. In this case the linguistic content is not changed. If we employ short-time SV, the speaker will not be verified for the substitution. If we consider this same possibility (word replacement with the same word) but with a voice clone, the recording will be verified. However, we recognize that prosodic differences, e.g. intonation, loudness, and stress in the replacement may impart a different meaning to the speech recording. We believe this would be difficult to carry out and may need further investigation.

5. CONCLUSIONS

Using existing techniques including digital signature, audio watermark, automatic speech recognition, and speaker verification, we have proposed a system to digitally sign a speech recording thereby verifying both speaker and linguistic content. This system may protect speech recordings against tampering with AI generated speech. The novelty of the system lies in combining these techniques where a cryptographic signature is used to verify linguistic content and speaker in a speech recording. This is possible due to a tight coupling between the signature and speech signal using an advanced watermark as well as accurate ASR and SV systems.

6. ACKNOWLEDGEMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant number 19/FFP/6775 and 13/RC/2077_P2.

7. REFERENCES

[1] P. R. Bevinamarad and M. Shirldonkar, "Audio forgery detection techniques: Present and past review," in *Proc. Int. Conf. Trends Electron. Informatics (ICOEI)*, 2020, pp. 613–618.

[2] X. Lin and X. Kang, "Exposing speech tampering via spectral phase analysis," *Dig. Sig. Process.*, vol. 60, pp. 63–74, 2017.

[3] C. Grigoras, "Digital audio recording analysis: The electric network frequency (ENF) criterion," *Int. J. Speech Language and The Law*, vol. 12, pp. 63–76, Jun. 2005.

[4] D. U. Leonzio, L. Cuccovillo, P. Bestagini, M. Marcon, P. Aichroth, and S. Tubaro, "Audio splicing detection and localization based on acquisition device traces," *IEEE Trans. Info. For. Sec.*, vol. 18, pp. 4157–4172, Jan. 2023.

[5] C. Kraetzer, A. Oermann, J. Dittmann, and A. Lang, "Digital audio forensics: a first practical evaluation on microphone and environment classification," in *Proc. Workshop on Multimedia & Secur. (MM&Sec)*. Assoc. Comput. Mach., 2007, pp. 63–74.

[6] Z.-F. Wang, J. Wang, C.-Y. Zeng, Q.-S. Min, Y. Tian, and M.-Z. Zuo, "Digital audio tampering detection based on ENF consistency," in *Proc. Int. Conf. Wavelet Anal. and Pattern Recognit. (ICWAPR)*, 2018, pp. 209–214.

[7] G. Ruggiero, E. Zovato, L. D. Caro, and V. Pollet, "Voice cloning: a multi-speaker text-to-speech synthesis approach based on transfer learning," *arXiv preprint arXiv:2102.05630*, 2021.

[8] R. Ratnawita, "Cybersecurity in the AI era measures deepfake threats and artificial intelligence-based attacks," *J. American Institute*, vol. 2, no. 2, pp. 180–189, 2025.

[9] S. Faziludeen, M. S. A. Sankar, P. L. De Leon, and U. Roedig, "Limitations of watermarking AI-generated speech using AudioSeal," in *Proc. IEEE Int. Conf. Trust, Privacy, and Security in Intell. Syst., and Applications (TPS-ISA)*, Nov. 2025.

[10] M. Steinebach and J. Dittmann, "Watermarking-based digital audio data authentication," *EURASIP J. on Adv. in Sig. Process.*, vol. 2003, no. 10, p. 252490, Sep. 2003.

[11] Coalition for Content Provenance and Authenticity (C2PA), "C2PA content credentials," https://c2pa.org/specifications/specifications/2.2/specs/C2PA_Specification.html, 2025, Accessed: Sep. 12, 2025.

[12] R. San Roman, P. Fernandez, H. Elsahar, A. Défossez, T. Furon, and T. Tran, "Proactive detection of voice cloning with localized watermarking," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2024.

[13] G. Chen, Y. Wu, S. Liu, T. Liu, X. Du, and F. Wei, "Wavmark: Watermarking for audio generation," *arXiv preprint arXiv:2308.12770*, 2023.

[14] L. Boney, A. H. Tewfik, and K. N. Hamdy, "Digital watermarks for audio signals," in *Proc. IEEE Int. Conf. Multimedia Comput. & Syst. (ICMCS)*, 1996, pp. 473–480.

[15] H. Liu, M. Guo, Z. Jiang, L. Wang, and N. Gong, "Audiomarkbench: Benchmarking robustness of audio watermarking," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 52 241–52 265, 2024.

[16] L. Hendrickson, "What are digital signatures? the complete guide," 2025, https://www.identity.com/what-are-digital-signatures , Accessed: Apr. 24, 2025.

[17] C. Hong, Z. Pei, Q. Wang, S. Yang, J. Yu, and C. Wang, "Quantum attack on RSA by D-Wave advantage: a first break of 80-bit RSA," *Sci. China Inf. Sci.*, vol. 68, no. 2, p. 129501, 2025.

[18] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Int. Conf. Machine Learning (ICML2023)*. PMLR, 2023, pp. 28 492–28 518.

[19] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors." in *Proc. Spoken Lang. Recogn. Workshop (Odyssey 2018)*, 2018, pp. 105–111.

[20] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak, P. A. Torres-Carrasquillo, and N. Dehak, "State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and speakers in the wild evaluations," *Comput. Speech & Lang.*, vol. 60, p. 101026, 2020.

[21] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 3830–3834.

[22] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database," *Nat. Inst. Stand. Technol. (NIST)*, vol. 107, p. 16, 1988.