A PUF Based Audio Fingerprint Based for Device Authentication and Tamper Location

Zhi Lu[1], Haochen Dou[1], Songfeng Lu[1,2], Xueming Tang[1(✉)], Junjun Wu[1], and Samir Mohammed Umran[1]

[1] Hubei Key Laboratory of Distributed System Security, Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
xmtang@hust.edu.cn

[2] Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518063, China

Abstract. As bioinformation authentication gains prominence, the significance of audio data in industries such as speech recognition intensifies, with audio storage becoming a pivotal concern for data protection. Existing audio tampering solutions fail to identify the producing device. This paper introduces an innovative method employing physical unclonable function (PUF) and audio features for identifying recording equipment and detecting tampered areas in judicial authentication within the Industrial Internet-of-Things (IIoT). The method comprises two components: the recording device, which generates an audio fingerprint using audio features and a PUF-determined random number seed, and the server, which registers, analyzes, and verifies the fingerprint. The unique, tamper-resistant PUF response is generated only when a server-provided challenge is initiated. The proposed audio fingerprint, evaluated using the Carioca 1 database and NXP LPC54S018-EVK-provided PUF functionality, enables varying tamper area identification accuracy and achieves 100% original device identification, resisting replay, cloning, and brute force attacks.

Keywords: Physical Unclonable Function · Voice Recoder Identification · Digital Audio Forensics · Landmark

1 Introduction

As information technology progresses incessantly, multimedia evidence has emerged as a crucial component in criminal prosecution. Audio, one of the most

This work is supported by the Major Research Plan of Hubei Province under Grant/Award No. 2023BAA027 and the Key Research & Development Plan of Hubei Province of China under Grant No. 2021BAA038 and the project of Science, Technology and Innovation Commission of Shenzhen Municipality of China under Grant No. JCYJ20210324120002006 and JSGG20210802153009028.

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 571, pp. 63–78, 2024.
https://doi.org/10.1007/978-3-031-56583-0_5

fundamental biological characteristics of humans, plays a vital role in court proceedings and other contexts. Over the past decade, recording devices have frequently served as vital evidence in verifying the authenticity and falsehood of particular crimes. The ongoing development of big data and the Internet of Things (IoT) has facilitated large-scale storage and utilization of audio data. However, the low cost of audio data acquisition has also increased the prevalence of audio-based attacks. Additionally, high-quality audio recording has contributed to the decreasing costs associated with audio tampering and theft. With the aid of manual technology, audio can be falsified and distorted, leading to miscarriages of justice in court. Consequently, audio tamper detection and recording equipment authentication are of utmost importance in the forthcoming digital era. In other words, it is essential to securely store audio data and ensure its protection, verification, and detection throughout its usage. At present, several studies have addressed image tamper detection and camera source recognition [6,13,20], but to the best of our knowledge, no work has effectively detected audio tampering while simultaneously identifying recording devices.

This paper proposes the utilization of audio features for identifying recording equipment and tampered regions in forensic authentication. The main contributions of our work include:

1. The proposed audio fingerprint can uniquely identify recording devices, enabling registration with a server. The server can link the audio requiring authentication and perform forgery detection. PUF-based SRAM is employed to generate the response of specific devices. The counter signifies the frequency of audio authentication, utilized for the projection transformation of audio landmark features to obtain audio fingerprints.
2. The generation of the audio fingerprint proposed in this paper depends on the number of authentications, audio data, and recording equipment, significantly enhancing the system's security. This method streamlines the recognition of individual recording devices, improving recognition speed and accuracy.
3. This work addresses the issue of impersonating original recording equipment in existing audio forensics. As the proposed audio fingerprint is device-dependent, authentication is undeniable. By authenticating the tagged audio, forgery attacks and re-defense attacks of recording equipment are eliminated.
4. The proposed method can dynamically adjust the effective detection region of tampering by modifying the threshold G and differentiate between malicious tampering and content-saving operations.

The remainder of this article is organized as follows: Sect. 2 discusses existing related work. Section 3 provides background information on Landmark and PUF in Shazam. Section 4 details the security model, system architecture, audio fingerprint generation, and server utilization for audio tamper detection and source recording device identification. Section 5 outlines the experimental setup and results, including data used, experiment preparation, testing, and security analysis. Section 6 provides a summary of the entire paper.

2 Related Work

In early audio forensics, the focus centered on analog tape recordings, gauging integrity via magnetic head switching transients, mechanical joints, overlapping recording signatures, and analog recorder fingerprints. Yet, contemporary audio data has largely shifted to digital format [2,11,12], posing challenges in verifying digital audio authenticity, establishing device-to-record links, spotting copies, and sequencing recorded events [8]. In recent times, diverse methods emerged to counter digital audio forgery, with [18] categorizing these detection techniques into three main groups:

1. ENF-based methods [1,2,7,12]: Electric network frequency (ENF) serves as a naturally embedded signature, which has emerged as a crucial development in audio evidence authentication. Owing to the natural regularity of ENF, audio clip insertions or deletions can create discontinuities in instantaneous phase information and produce abnormal changes in the registered ENF [12].
2. Post-processing distortion-based methods: [8,16] typically utilize Mel frequency cepstral coefficients or the statistical properties of modified DCT coefficients to identify traces of double compression left by operations.
3. Acoustic environment-based methods: By extracting inconsistencies in acoustic environment features from audio files, [10] detect tampered positions and splicing. Researchers have verified audio recordings by examining audio reverb introduced during recording due to sound persistence after the sound source's termination.

These methods are considered passive audio forgery detection techniques [14]. In contrast, active detection mechanisms involve inserting or attaching additional information to ensure security and integrity when creating target digital audio. Active audio forgery detection methods are primarily divided into two categories:

1. Watermark-based methods: These methods enhance audio integrity and security by embedding watermarks, which can be extracted from the audio after network transmission. An intact watermark in the audio content signifies its origin from an authenticated source, distinguishing it from recordings or playback devices [4].
2. Digital signature-based methods: Employing the user's private key, digital signatures encrypt information. The recipient decrypts the received content using the sender's public key, maintaining integrity and authenticity during transmission over insecure public channels [3].

3 Preliminaries

3.1 Landmark

We use the 'Landmark' audio fingerprint generated by Shazam's hashing process, consisting of three modules [15]:

1. **Robust Constellations:** Shazam identifies spectral maxima indicating points where a time-frequency region's value surpasses neighbors. A density criterion ensures robustness by evenly distributing selected maxima, simplifying the spectral graph into sparse coordinates.
2. **Fast Combinatorial Hashing:** Shazam swiftly indexes constellations by hashing two time-frequency points. Anchor points, tied to target zones, are sequentially combined with target area points, yielding two frequencies and a time difference. Each fingerprint, stored as an INT, includes the time difference from the file's start to the anchor point, excluding absolute time.
3. **Searching and Scoring:** Extracted hashes match database hashes, creating time pairs for each match. These pairs form a scatter plot. When test audio matches database audio, similar time pairs form a diagonal line: $t_k^{'} = t_k + offset$, where $t_k^{'}$ is the source audio's matched hash time, and t_k is the test audio's matched hash time.

3.2 Physical Unclonable Function

A Physical Unclonable Function (PUF) is a unique hardware response/challenge function [9] that exploits inherent random variations introduced by manufacturing processes to form secret keys on-the-fly, analogous to a unique fingerprint [5]. PUFs can be utilized with minimal hardware investment and are unclonable due to their unpredictable challenge-response behavior. This study uses Static Random-Access Memory (SRAM) PUF as the response generator, which is based on the behavior of SRAM cells in memory. Each cell's state after power-up is determined by random differences in threshold voltages, yielding a unique binary pattern. The response of SRAM PUF depends on the threshold voltage error, making it challenging to clone the PUF's challenge-response behavior.

4 Proposed Approach

4.1 Security Model

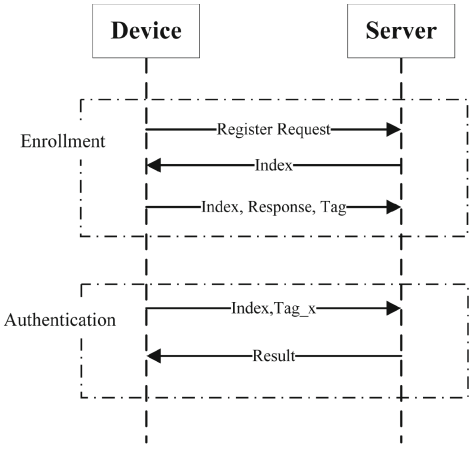


Fig. 1. Framework of proposed authentication process between server and device.

We assume that during the enrollment phase, communication between parties is secure, relying on existing security protocols such as SSL and TLS. Each audio enrollment request receives a unique index, and the server obtains a device response. In the authentication phase, the channel between the device and server is unreliable, allowing adversaries to manipulate, eavesdrop, and replay messages. Adversaries can request authentication from the server using past or predicted messages and attempt to impersonate the original device using a sham device with enrolled original audio. The original device can also submit tampered audio to the server. Upon receiving tags, the server verifies their correctness, integrity, time sequence, and origin. Consequently, our scheme ensures verifiability and resists replay and cloning attacks.

4.2 System Architecture

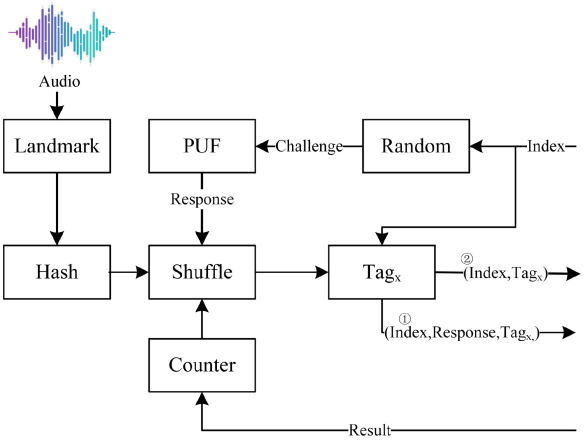


Fig. 2. Framework of proposed Device.

Figure 1 illustrates a system involving two entities: the client authentication device and the server. When a device enrolls audio, it sends an enrollment request to the server, which returns a unique index. The device generates a Tag and a PUF Response for reshuffling, sends the index, Tag, and Response to the server, which securely stores them in a database. To verify audio legitimacy, the device sends the index and Tag to the server for integrity and identity authenticity checks. The server sends the result to the device and increments the Counter, ensuring distinct tags each time. The client-side authentication device framework in Fig. 2 includes an SRAM PUF, adding device-specific "genes" to the generated audio fingerprint. The device uses the server-provided index as a Random seed, projecting the index onto PUF's required mapping space. The PUF receives the challenge and outputs the corresponding Response. The device owner extracts audio landmarks, hashes them, and shuffles the hash with the PUF's Response and a Counter representing authentication requests. During enrollment, the device outputs the index, Response, and Tag$_x$ (Fig. 2's ①); for authentication, it outputs the index and Tag$_x$ (Fig. 2's ②). The device retrieves the result and increments the Counter. Figure 2 shows the device shuffling the landmark hash list into a Tag using the Response and Counter of authentication requests.

We adapted the Knuth shuffle for our framework (Algorithm 1). The shuffle function appends the Counter to the hash list's tail and sets x to the Counter's number. First, it XORs every hash in the list, incrementing x. The XOR hash key is generated by a Random function with the input of the x_{th} response item. Second, it swaps hash items with k generated by the Random function. As the

Response is a PUF output, adversaries cannot imitate this shuffle process, which also acts as symmetric encryption. The Response, stored in the server during enrollment, serves as the encryption key, ensuring resistance to replay attacks.

Algorithm 1. Shuffle

Input: $H = h_0, ..., h_{n-1}$, $Response = r_0, ..., r_{m-1}$, $Counter$
Output: Tag

function SHUFFLE($H, Response, Counter$)
  $i \leftarrow 0$
  $H \leftarrow$ append $Counter$ to the end of the H
  $x \leftarrow Counter$
  for $i \leftarrow 0$ to n **do**
    $Random.seed(Response[x \bmod m])$
    $b \leftarrow Random$
    $H[i] \leftarrow H[i]$ XOR b
    $x \leftarrow x + 1 \bmod m$
  end for
  for $i \leftarrow 0$ to n **do**
    $Random.seed(Response[x \bmod m])$
    $k \leftarrow random$
    $swap(H[i], H[k])$
    $x \leftarrow x + 1 \bmod m$
  end for
  Tag $= H$
  return Tag
end function

The server, a secure service provider, generates an index for each device and performs information authentication. As shown in Fig. 3, upon enrollment request, the server generates a unique index, sends it to the client, and stores the hash, Response, index, Counter, and times in the database. The hash is reshuffled from the audio Tag, and the Counter represents authenticated audio requests and consecutive failed requests. In the authentication phase, the server receives the index and unverified Tag, reshuffles the tags using the stored Response and Counter, and obtains unverified hash and Counter. Comparing the two counters, if they differ, the request is deemed tampered or a replay attack, failing authentication. If the Counter values match, the server uses Shazam's Searching and Scoring to compare the hashes and identify tampered audio sections. A threshold value G controls the tampered area, as landmark values near the tampered area change. The tampered area is always smaller than the detected area. A limit G bounds the largest detected area; if exceeded, tamper location detection fails. The limit area, given t_o as the stored audio time and t_t as the time of the audio awaiting authentication, is:

$$log_{10}^{\frac{3}{32}G^2+\frac{5}{8}} min(t_0, t_t) + log_2^{\frac{10}{G}} |t_0 - t_t| \tag{1}$$

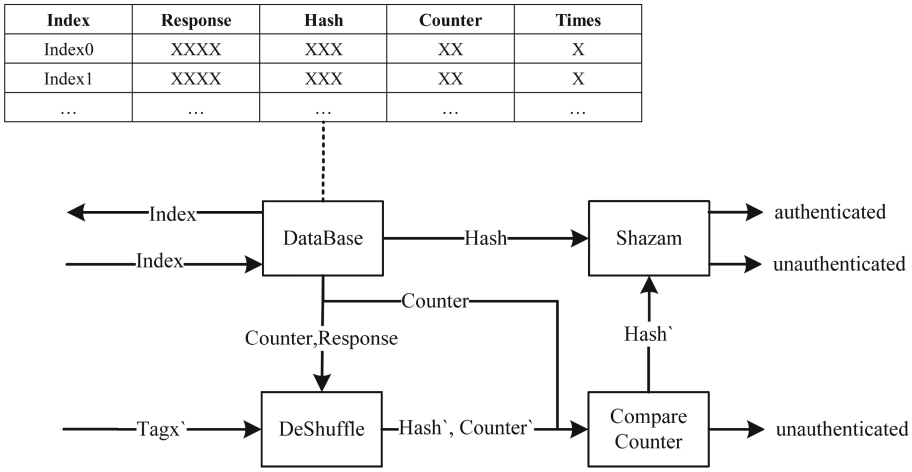


Fig. 3. Framework of proposed Server.

On the server-side, the server makes reshuffle to get the landmark hash and the Counter to be authenticated with the input of Tag, Response, and the Counter stored in the server's database. The reshuffle function's process is the inverse of shuffle. It should be considered that the initial x is not Counter but Counter plus $2n-1$, n representing the length of Tag.

5 Experiments and Results

5.1 Database

In this section, we utilize the publicly available Carioca 1 database [17] as the test database, allowing for direct comparisons with methods from [1,2,7,11], and [12]. The Carioca 1 database contains 100 original audio files, 50 from female speakers and 50 from male speakers. Additionally, there are 100 tampered versions of these audio files, with 50 containing inserted audio clips and 50 featuring deleted audio clips, distributed evenly between male and female speakers. Each audio file undergoes a single cut or insertion. The timings (sample indices) and extents of each edit are carefully documented. Edits are performed in sound-inactive portions of the signal. Upon analyzing the Carioca 1 audio files, the Signal-to-Noise Ratio (SNR) distribution in the active voice areas ranges from 16 dB to 30 dB, with an average of 22.3 dB.

5.2 Experimental Setup

In our experiments, during the landmark extraction process, we set the parameters as follows: $N_{FFT} = 512$, $N_{HOP} = 256$. We limit the maximum number of local maxima per frame to 5 and restrict the number of pairs derived from each

peak to less than 3. The density of landmarks found is represented by $density = 20$, and the spectrogram enhancement has an $HPF\ POLE = 0.98$. We set the target sampling rate at 44.1 kHz, matching the sampling rate of the audio files in the Carioca 1 database. To reduce the framing effect, we extract the hash value from two shifted waveforms ($shifts = 2$). The hash is generated from the Landmark feature and has the following form: $landmark = (time, bin1, bin2, \Delta time)$. We perform fast combinatorial hashing on the landmark, combining two time-frequency points to form a fingerprint hash $= [time, h_o]$, where $h_o = h_1|h_2|h_3$.

$$\begin{aligned} h_1 &= (bin1\&255) \ll 12 \\ h_2 &= (bin2 - bin1)\&63) \ll 6 \\ h_3 &= \Delta time\&63 \end{aligned} \tag{2}$$

Then, we combine the $time$ and h_o together by: $H = time \ll 32 + h_0$. By doing the inverse operation to the H, we can get $time = H \gg 32, h_o = H\&((1 \ll 32) - 1)$. The 32-bit hash, H, is derived from the 256-bit PUF Response, which is divided into 32 groups, each containing 8 bits. The PUF is provided by the LPC54S018-EVK, an NXP Development Board with LPC54S0xx MCU as shown in Fig. 4. This MCU utilizes dedicated SRAM to generate a PUF root key.

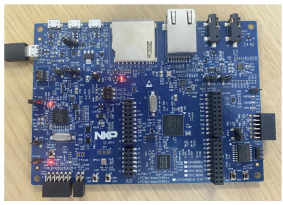
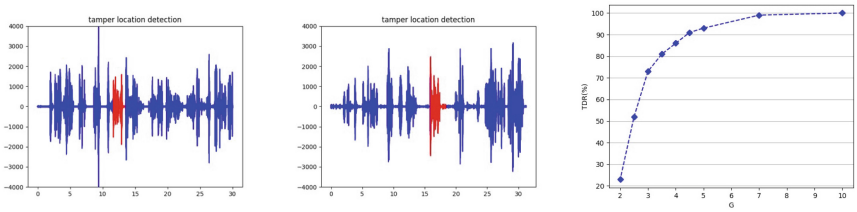


Fig. 4. NXP LPC54S018-EVK Development Board.

The proposed edit detection method's performance, measured in tamper detection rate (TDR), is evaluated under four conditions: original audio recordings from the database; compressed database audio; amplitude clipping applied to database audio; and additional noise added to the database audio. It is important to note that edit detection is divided into two situations: detecting whether the audio has been tampered with and determining the location of the tampering. These situations are controlled by the parameter G, a threshold determining the accuracy of tampered location detection. Smaller values of G yield more accurate tamper localization. At its maximum value, G allows the proposed method to function as an edit detector. The performance of the proposed identification method is assessed under three conditions, with each condition selecting one audio, device, or challenge as the independent variable. Concurrently, the analysis of the audio database reveals that the signal-to-noise ratio of the original audio background noise estimation does not exceed 30 dB.

5.3 Tamper Location Detection Test



(a) cut area (b) insert area (c) insert area

Fig. 5. Sample of original audio tamper detection

Original Recording of Audio. Figure 5(a) and (b) showcase samples identifying the cut area and insert area. Contaminated areas are in red and normal areas are in blue Fig. 5(c) depicts a line chart of TDR with different values of G. As G decreases, TDR declines. When $G = 10$, the method serves as a tamper detector; when $G = 2$, the method succeeds only if the tampered area size is nearly equal to the allowed detection area size. Given that the server possesses audio information, a G value of 10 consistently detects tampering. We prefer $G = 4.5$, as it achieves over 90% accuracy, and the tampered area size is roughly half the allowed detection area size.

Compression of Database Audio. We employ the same method as [12] to test our proposed method's performance under MP3 compressed audio signals. To assess the robustness of our method under compression, we converted the Carioca 1 database audio files to MP3 format and recoded the entire database at a sample rate of 2250 Hz and a set of $Br = [16, 32, 64, 128]$ kbps, similar to [12], to compress the bit rate.

Table 1 demonstrates that accuracy decreases with decreasing G or BR values. When $G = 10$, the tamper detection rate is 100%, and compression does not impact performance. Our proposed method's performance is only greater than 93% when $G \geq 7$, permitting an allowed detection area nearly larger than half the audio. When $G = 4.5$, the TDR is reduced by 1%–8%.

Table 1. TDR of Compress tests with different kpbs and G

G	16BR	32BR	64BR	128BR
10	100%	100%	100%	100%
7	93%	97%	97%	98%
5	83%	87%	89%	90%
4.5	76%	80%	86%	87%
4	66%	72%	79%	80%
3.5	59%	61%	66%	69%
3	40%	54%	56%	58%
2.5	21%	28%	30%	30%
2	8%	11%	12%	12%

Apply Amplitude Clipping. To achieve amplitude clipping of the database signal, we use the same VAD method employed in [12]. The saturation level (SL) represents the percentage of active speech samples clipped to a maximum, corresponding to a specified number. The set of specified $SL = [0, 0.2, 0.5, 1, 2, 4]\%$ is the same as in [12].

Table 2. TDR of Amplitude tests with different SL and G

G	0.2SL	0.5SL	1SL	2SL	4SL
10	100%	100%	100%	100%	100%
7	98%	97%	97%	97%	96%
5	91%	91%	89%	87%	80%
4.5	89%	87%	83%	80%	71%
4	79%	73%	71%	69%	64%
3.5	66%	64%	59%	58%	52%
3	58%	52%	48%	43%	38%
2.5	33%	27%	24%	21%	19%
2	13%	12%	10%	6%	7%

Table 2 demonstrates that accuracy decreases with the decrease of G or BR values. When $G = 10$, the tamper detection rate is 100%, and the compression operation does not affect the performance. The performance of our proposed method is only greater than 96% when $G \geq 7$, allowing for a detection area nearly larger than half the audio. When $G = 4.5$, the TDR is affected and decreased by 2%–20%. The performance of tamper location detection decreases significantly when the amplitude clipping is at $4SL$.

Additional Noise. To calculate the audio signal's power P_s and the power of generating noise P_{n1} (assuming the signal length is both N):

$$P_s = \frac{\sum_{i=1}^{N}(x_i)^2}{N}, P_{n1} = \frac{\sum_{i=1}^{N}(n_i)^2}{N} \tag{3}$$

It is important to note that the noise we currently generate, which is the SNR with an audio signal, does not match the $SNR\ dB$ we require. In other words, we need to multiply all the noise data by a number k, ensuring its power meets our requirements. Where P_n is the power of noise we ultimately need:

$$SNRdB = 10 \cdot log_{10}\frac{P_s}{P_n} \tag{4}$$

By deducing and sorting out Eqs. (3) and (4):

$$k = \frac{P_s}{10^{\frac{SNR}{10}} \cdot P_{n1}} \tag{5}$$

Table 3 displays the results of our proposed method's performance. The performance of $G = 4.5$ has decreased by 6%–23%.

Table 3. TDR of Noisy tests withe different SNR and G

G	5SNR	10SNR	15SNR	20SNR	25SNR	30SNR
10	100%	100%	100%	100%	100%	100%
7	96%	96%	96%	96%	96%	97%
5	72%	80%	84%	86%	87%	87%
4.5	68%	73%	78%	82%	83%	84%
4	52%	63%	70%	72%	75%	75%
3.5	41%	51%	58%	63%	64%	64%
3	26%	37%	46%	49%	52%	55%
2.5	13%	18%	22%	26%	29%	30%
2	4%	4%	8%	10%	10%	11%

5.4 Source Audio Identification

Our proposed method ensures that the Tag to be validated can only be reshuffled successfully by the device that generated the audio. The feature unique to the original device is mixed into registration information. The server has the ability to detect whether the device that sent the audio for detection is the device that produced the audio. On this basis, it can also detect tampering and tampering location, which has been tested above. There are three independent variables: device, audio, and challenge. We perform three tests to verify the identification ability, similar to [19].

Test 1: The degree of differentiation of the Tag generated by using different PUF challenges with the same audio and the same device.
Test 2: The degree of differentiation of PUF responses generated by the same PUF challenge input into different devices.
Test 3: The degree to which the PUF response generated by the same PUF challenge input to the same device is distinguishable from the Tag generated by the combination of different audio frequencies.

We use two LPC54S018-EVK development boards, 100 authentic audio files belonging to the Carioca 1 database, and 100 indexes that represent the device challenges. The devices are labeled as d_1 and d_2, the audio files are labeled as a1 to a100, and the challenges are labeled as 1 to 100. For each test, there are a total of $2 \times 100 \times 100 = 20,000$ test data points.

Table 4. EAR of Source audio Identification with three independent variables

EAR	challenge	device	audio
condition1	0%	0%	0%
condition2	0%	0%	0%

Here we chose the Tag generated by d_1 and index = 1 to be the enrollment information. The landmark of the audio, Response of PUF, and the Counter of times the audio has been authentic generate the Tag. We assume two conditions: 1. Each time the test audio is authenticated, the Counter defaults to be 1, and the Counter of the audio is known to the attacker. 2. The Counter increases with time, which means that the Counter ranges from 1 to 100. Meanwhile, the adversary does not know the value of the count. Table 4 shows the error acceptance rate (EAR) of Test 1, Test 2, and Test 3. The result of the challenge test is 0. This is because the Response of the PUF does not have a collision. The Response here is 256 bits, and the space of Response is 2^{256}. The equation for the probability of a collision is:

$$1 - e^{\frac{-k(k-1)}{2N}} \approx 0, N = 2^{256}, k = 20000$$

The result of the device test is 0. This is because of the PUF's feature, which ensures the device's unique unclonability. The result of the audio test is 0. This is due to the feature of the landmark. As the length of the feature depends on the peaks and length of the audio, it means that two audios rarely have the exact same landmark unless they are the same audio.

5.5 Comparison

Our proposed method is compared with the existing audio tamper detection methods in recent years. The comparison methods all use the same database,

Carioca 1. The comparison is carried out from the following four aspects: robustness, tamper detection, tamper location detection, and device authentication, as shown in Table 5. For robustness, as different methods have a significant difference between their theories, the methods of [2,7,12] use ENF variations with different technology. Meanwhile, [12] uses SVM and [7] uses an autoregressive model. Tables 1, 2, 3 show the compress, amplitude, and noise, respectively. One significant difference between our approach and theirs is that our method is built with known basic audio information, and it can determine whether the device verifying the audio is the original device.

Table 5. Comparison with existing methods

Method	Robustness			Tamper		Authentication
	Compress	Noisy	Amplitude	Detection	Location	
[12]	YES	YES	YES	YES	YES	NO
[7]	YES	YES	NO	YES	NO	NO
[2]	NO	YES	YES	YES	YES	NO
proposed method	**YES**	**YES**	**YES**	**YES**	**YES**	**YES**

5.6 Security Analysis

In this section, we describe details about how the proposed system defends against the adversary's attacks mentioned above.

The **replay attack** in our scenario is that during the authentication, the adversary could monitor the communication channel and get the index and tags. Then, the adversary could impersonate the client to perform the authentication. In order to defend against the replay attack, we incorporate a counter in our design. During the shuffle process, the input of the mask code is the counter value, making the mask code one-time use. After accepting the tags, the server reshuffles the tags with the stored counter and updates it. Thus, the tags captured by the adversary are useless and cannot be verified the next time.

The **cloning attack** in our scenario refers to the duplication of another device that gets the same input as the claimed device in order to generate the same Response. After capturing the index in the authentication process and assuming the adversary has the ability to count the authentication times, typically, if we adopt a one-way hash function like SHA, the Response is absolutely the same with the same input. What is worse, the attacker could generate the same tags and index itself and complete the authentication process. However, PUF makes such an attack infeasible even with the same input. Because of the physical characteristics, the whole manufacturing process is uncontrollable, which is entirely different from cryptography methods applied in hash algorithms. That is, our system can link each registered audio to the specified device due to the application of PUF.

The **brute force attack** in our scenario occurs during the authentication phase. Assume the adversary has the past or predicted Tag of this audio. The

time in the server's database provides an additional layer of security against opponents who use random input to brute force a break on the server. When the number of authentication failures is greater than a threshold value, the server rejects the request for this index until the administrator resets it.

6 Conclusion

The method we proposed is the first of its kind to utilize PUF as a device gene for generating audio fingerprints. The generated audio fingerprints enable tamper detection and tamper location under conditions of anti-compression, shearing, and noise. The detection of the original audio device can be achieved to prevent device cloning and counterfeiting. Additionally, defenses against possible brute force and replay attacks are implemented. This is achieved by using landmarks as the audio feature, which, in conjunction with PUF and Counter, generates audio fingerprints. Due to PUF's unclonability and unpredictability, it is virtually impossible for an attacker to predict the correct audio fingerprint. This innovative approach offers a robust and secure method for verifying the integrity and authenticity of audio files, as well as identifying the devices responsible for their creation, effectively protecting against various types of attacks.

References

1. Esquef, P.A.A., Apolinário, J.A., Biscainho, L.W.P.: Improved edit detection in speech via ENF patterns. In: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6 (2015). https://doi.org/10.1109/WIFS.2015.7368585
2. Esquef, P.A.A., Apolinário, J.A., Biscainho, L.W.P.: Edit detection in speech recordings via instantaneous electric network frequency variations. IEEE Trans. Inf. Forensics Secur. **9**(12), 2314–2326 (2014). https://doi.org/10.1109/TIFS.2014.2363524
3. Forouzan, B.: Cryptography and Network Security. McGraw-Hill, New York (2007)
4. Garlapati, B.M., Kakkirala, K.R.: Malicious audio source detection using audio watermarking. In: 2015 Asia Pacific Conference on Multimedia and Broadcasting, pp. 1–5 (2015). https://doi.org/10.1109/APMediaCast.2015.7210288
5. Herder, C., Ren, L., van Dijk, M., Yu, M.D., Devadas, S.: Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. IEEE Trans. Dependable Secure Comput. **14**(1), 65–82 (2017). https://doi.org/10.1109/TDSC.2016.2536609
6. Hu, W., Chang, C.H., Sengupta, A., Bhunia, S., Kastner, R., Li, H.: An overview of hardware security and trust: threats, countermeasures, and design tools. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **40**(6), 1010–1038 (2021). https://doi.org/10.1109/TCAD.2020.3047976
7. Lin, X., Kang, X.: Supervised audio tampering detection using an autoregressive model. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2142–2146 (2017). https://doi.org/10.1109/ICASSP.2017.7952535

8. Liu, Q., Sung, A.H., Qiao, M.: Detection of double MP3 compression. Cogn. Comput. **2**, 291–296 (2010). https://doi.org/10.1007/s12559-010-9045-4
9. Maes, R.: Physically Unclonable Functions: Constructions, Properties and Applications. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41395-7
10. Malik, H.: Acoustic environment identification and its applications to audio forensics. IEEE Trans. Inf. Forensics Secur. **8**(11), 1827–1837 (2013). https://doi.org/10.1109/TIFS.2013.2280888
11. Nicolalde Rodriguez, D.P., Apolinario, J.A., Biscainho, L.W.P.: Audio authenticity: detecting ENF discontinuity with high precision phase analysis. IEEE Trans. Inf. Forensics Secur. **5**(3), 534–543 (2010). https://doi.org/10.1109/TIFS.2010.2051270
12. Reis, P.M.G.I., Lustosa da Costa, J.P.C., Miranda, R.K., Del Galdo, G.: ESPRIT-Hilbert-based audio tampering detection with SVM classifier for forensic analysis via electrical network frequency. IEEE Trans. Inf. Forensics Secur. **12**(4), 853–864 (2017). https://doi.org/10.1109/TIFS.2016.2636095
13. Shi, J., Wang, G., Su, M., Liu, X.: Effective medical image copy-move forgery localization based on texture descriptor. In: Goel, S., Gladyshev, P., Johnson, D., Pourzandi, M., Majumdar, S. (eds.) ICDF2C 2020. LNICST, vol. 351, pp. 62–77. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68734-2_4
14. Teerakanok, S., Uehara, T.: Digital media tampering detection techniques: an overview. In: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 170–174 (2017). https://doi.org/10.1109/COMPSAC.2017.109
15. Wang, A.L.: An industrial-strength audio search algorithm. In: Choudhury, S., Manus, S. (eds.) 4th Symposium Conference on Music Information Retrieval, ISMIR 2003, pp. 7–13. The International Society for Music Information Retrieval (2003). http://www.ismir.net. http://www.ee.columbia.edu/~dpwe/papers/Wang03-shazam.pdf
16. Yang, R., Shi, Y.Q., Huang, J.: Detecting double compression of audio signal. In: Memon, N.D., Dittmann, J., Alattar, A.M., Delp, E.J., III. (eds.) Media Forensics and Security II, vol. 7541, pp. 200–209. International Society for Optics and Photonics, SPIE (2010). https://doi.org/10.1117/12.838695
17. Zeng, J., et al.: Audio recorder forensic identification in 21 audio recorders. In: 2015 IEEE International Conference on Progress in Informatics and Computing (PIC), pp. 153–157 (2015). https://doi.org/10.1109/PIC.2015.7489828
18. Zhao, H., Chen, Y., Wang, R., Malik, H.: Anti-forensics of environmental-signature-based audio splicing detection and its countermeasure via rich-features classification. IEEE Trans. Inf. Forensics Secur. **11**(7), 1603–1617 (2016). https://doi.org/10.1109/TIFS.2016.2543205
19. Zheng, Y., Cao, Y., Chang, C.H.: A PUF-based data-device hash for tampered image detection and source camera identification. IEEE Trans. Inf. Forensics Secur. **15**, 620–634 (2020). https://doi.org/10.1109/TIFS.2019.2926777
20. Zuo, H., Li, Q., Zheng, H., Yang, Y., Zhao, X.: An optically-reconfigurable PUF based on logarithmic photoreceptor of CMOS dynamic vision sensors. IEEE Trans. Electron Devices **69**(9), 5395–5398 (2022). https://doi.org/10.1109/TED.2022.3191628