L. Hernández Acosta, A. Reinhardt, T. Müller, and D. Reinhardt. Beyond Wake Words: Advancing Smart Speaker Protection with Continuous Authentication and Local Profiles. Proceedings of the 33rd International Conference on Computer Communications and Networks (ICCCN), 2024.

Beyond Wake Words: Advancing Smart Speaker Protection with Continuous Authentication and Local Profiles

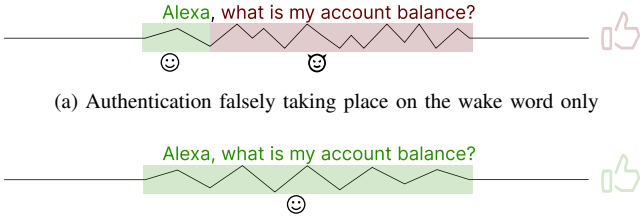
Luca Hernández Acosta*, Andreas Reinhardt†, Tobias Müller*, Delphine Reinhardt*
*Computer Security and Privacy (CSP), Institute of Computer Science, University of Göttingen, Göttingen, Germany
hernandez@cs.uni-goettingen.de, tobiasmueller@posteo.de, reinhardt@cs.uni-goettingen.de
†Dept. of Informatics, TU Clausthal, Clausthal-Zellerfeld, Germany
reinhardt@ieee.org

***Abstract*—Voice assistants, as provided by smart speakers, have become ubiquitous. Current authentication methods in these systems, however, rely on wake words, posing a risk due to the susceptibility to replay attacks. Additionally, user data stored on servers could expose sensitive information. This study suggests an approach to improve user authentication and profile management in smart speakers, reducing risks tied to external data processing and storage. We propose a two-fold solution for continuous user authentication and local user profiles. This approach prevents unauthorized access to sensitive data and grants users access to their local recordings. Our method differs from current practices in two ways: (1) It authenticates users based on complete voice commands, reducing the risk of replayed wake word attacks, and (2) it operates locally, avoiding the transfer of sensitive data to external servers. We offer a proof-of-concept with *Alexa Voice Service* (AVS) integration and a thorough evaluation using voice datasets and a study with 17 participants. We tested our approach under various conditions, including accents, background noise, and muffled speech. Legitimate users are identified with 93% precision, 95% recall, 94% F1-score, and 99% accuracy, while illegitimate users are recognized with 99% accuracy across these metrics.**

***Index Terms*—smart speakers, voice assistants, voice authentication, replay attacks, local profiles, privacy**

I. Introduction

The widespread use of voice-controlled devices in both personal and professional environments has been transformative. As the adoption of smart homes increases, people are frequently using voice assistants on devices like smart speakers, smartphones, and smartwatches. However, this convenience raises privacy concerns, as these assistants continuously listen for activation commands ("wake words") and send subsequent audio to remote servers for processing, which includes command interpretation through *Natural Language Processing* (NLP) and speaker verification. Smart speakers primarily authenticate users based on the wake word, a method that poses significant security risks. This system is vulnerable to replay attacks, where a malicious individual can replay a recording of the legitimate user uttering the wake word, followed by unauthorized commands. This could allow unauthorized access to sensitive information such as personal emails, calendar events, financial details, or shopping histories



(a) Authentication falsely taking place on the wake word only

(b) Authentication taking place on the full voice command

Fig. 1. Comparison of current and proposed authentication methods

(see Fig. 1a). To prevent replay attacks, different techniques have been proposed to test the speaker's liveness [1–7], i.e., if a real and living user is actually saying the voice command. These techniques, however, have several limitations: (1) they require additional hardware, such as specialized sensors or devices to measure oral airflow or ear canal pressure, making them less accessible and more costly to implement [1, 3]; (2) they may not effectively distinguish between a live speaker and high-quality recorded audio in environments with variable acoustic characteristics or when sophisticated spoofing techniques are used [3, 6, 7]; and (3) they do not offer protection if both the genuine user and the honest-but-curious user are together, as these methods might not be able to differentiate between simultaneous inputs from multiple users or prevent unauthorized access when a legitimate user is coerced into providing a voice command. In such cases, the latter user can surprise the genuine one by interjecting and pronouncing another command to get access to private content. While this scenario requires the co-presence of both users and precise timing by the honest-but-curious user, it remains a feasible risk. To address these limitations, we therefore propose that the whole voice command and not only the wake word should be analyzed and authenticated in a text-independent manner, as visualized in Fig. 1b. Moreover, using the traditional approach of wake word detection for activation and subsequent data transmission to service providers, a significant amount of user data is collected by the service provider. Not only the content of these data may reveal sensitive information about

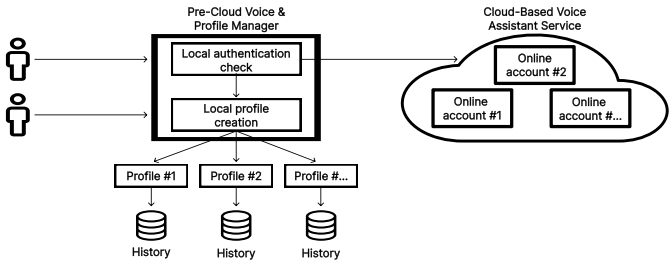


Fig. 2. System overview: Local voice checker and local profile creation

the users, but also their voice [8], ranging from emotional state to health issues [9]. Furthermore, voice recordings were not only accessed without user consent in the past, but also sold or publicly disclosed [10, 11]. Such unwanted disclosures are not only limited to service providers. Indeed, they can also happen in common multi-user environments. In this setting and in absence of effective voice-based authentication, secondary users can easily access sensitive information tied to the primary user's account. Conversely, primary users can review the secondary users' interaction history; regardless if registered or not. While many voice assistants offer voice authentication, their efficacy in safeguarding privacy varies significantly among providers. Often, primary users who initially set up the device retain the ability to access the interaction history of other authenticated secondary users. This is the case for popular voice assistant platforms, including those developed by major technology companies like Amazon (Alexa). To access their interaction history, secondary users must create an online account, providing personal details like email and residence. This process helps manufacturers build even more detailed user profiles [12]. To prevent it, we propose a different approach with which local user profiles are created based on features extracted from complete voice commands. After their creation, the respective users' past interactions are linked to them. By doing so, only the concerned user gets access to her history and she does not need to register with the service provider.

Combining both aspects, our goal is hence to investigate the feasibility of performing continuous voice authentication and profile creation, both locally on the smart speaker, as shown in Fig. 2. By doing so, we aim to shift from existing systems where all voice interactions are attributed to an online account, to one where individual users' interactions are locally pre-filtered, managed, and identified on the device (i.e., a smart speaker in our case) before being forwarded to the provider's cloud service (e.g., *Alexa Voice Service* (AVS)) for actual voice assistance and other online-only features. Note that additional local privacy-protecting measures could be applied to, e.g., remove voice characteristics or sensitive words [8] according to the individual users' privacy preferences before their further transmission. To reach our goal, we make the following contributions.

1) Introduction of a continuous authentication method, verifying entire voice commands using an LSTM network with MFCC features. Proof-of-concept integrated with AVS on a Raspberry Pi.
2) Proposal of a novel approach for local user profiles and interaction histories, eliminating manufacturer registration needs.
3) Evaluation of continuous authentication scheme, achieving over 90% for each metric precision, recall, F1-score, and accuracy, ensuring high performance in recognizing legitimate users.
4) Analysis of replay attack detection performance, achieving 99% precision, recall, F1-score, and accuracy.
5) Performance evaluation in realistic scenarios: analyzing accents, background noise effects, and muffled voices.
6) Conducted user study to assess command duration, languages, and multi-user environments. Results show system's ability to manage multiple users and languages with decreased precision in larger groups.
7) Feasibility evaluation showing average verification time and consistent system resource usage across speakers, confirming practical viability without significant overhead.

The remainder of this paper is structured as follows. In Sec. II, we provide an overview of related work. We introduce our concept in Sec. III and give insights about our system model and the adopted scenario. We further discuss the design of our local profiles and implementation of the continuous authentication mechanisms in Sec. IV. Sec. V describes the evaluation setup, including a discussion of the dataset and the required preprocessing steps as well as the evaluation metrics. In Sec. VI, we delve into our evaluation results. We discuss these results and give an outlook in Sec. VII, before drawing conclusions in Sec VIII.

II. Related Work

The domain of voice processing and authentication is both expansive and interdisciplinary. Our research spans multiple areas, including privacy in voice data storage, local processing for voice recognition, and continuous authentication beyond wake words. Each of these areas presents its own set of challenges and opportunities, which we present in what follows.

A. Privacy Concerns in Voice Data Storage

One of the most pressing issues in modern voice recognition systems is the storage and handling of voice data. Currently, most commercial smart speakers store these data on cloud servers owned by the manufacturer [13, 14]. This not only poses a risk to user privacy, but also makes the system vulnerable to data breaches [10, 11]. However, it has been noted that there is a demand for a history of voice interactions to be accessible, indicating a common desire among users to revisit past requests or actions [15]. Our research aims to shift this paradigm by storing voice recordings locally, thereby giving users full control over their own data.

B. Local Processing for Speech and Voice Recognition

Voice recognition systems have traditionally relied on cloud-based solutions [16]. However, there is a growing interest in

local processing to reduce latency and enhance privacy by keeping sensitive voice data on the device itself [17, 18]. Research in this area focuses on the feasibility and efficiency of implementing speech and voice recognition algorithms locally. We hence contribute to this line of research by presenting a novel approach that centers on local authentication and profile management, enabling the storage of user audio data directly on the device.

C. Beyond Wake Words: Continuous Authentication

Traditional voice recognition systems often rely on a wake word for initial authentication [19]. However, this approach is limited to text-dependent speaker recognition [19]. The need for more secure and continuous authentication has led to research focusing on text-independent speaker recognition [20]. This involves analyzing various features of the speaker's voice in real-time, beyond just the wake word, to ensure ongoing authentication and prevent replay attacks. Advancements in liveness detection specifically address replay attacks, often targeting full commands, by employing methods like sound field dynamics, ear canal pressure, and oral airflow detection [1–7]. These techniques ensure the presence of a live speaker, enhancing security against such attacks. However, limitations related to user acceptability, environmental variability, and the need for additional hardware highlight the ongoing necessity for a comprehensive solution that extends beyond wake word authentication for a more secure, continuous, and adaptable voice authentication.

D. Speaker Recognition

In recent years, the speaker recognition field has transitioned from traditional methods like *Gaussian Mixture Models* (GMMs) to deep learning approaches, particularly *Deep Neural Networks* (DNNs) [21]. These methods, including *Convolutional Neural Networks* (CNNs), *Long Short-Term Memory* (LSTM) networks, and advanced architectures like BERT and Transformer, have become the current state of the art, offering improved performance and robustness in challenging environments [21]. LSTM networks, known for their ability to capture temporal dependencies, align well with tasks involving *Mel-Frequency Cepstral Coefficients* (MFCCs) for speaker recognition, where understanding time-based voice patterns is crucial for accurate identification and verification [22, 23]. Given the ongoing technological advancements, we utilize LSTM networks in our speaker recognition system for their capability to capture temporal speech patterns.

III. Concept

As discussed in Sec. I, reliable user authentication for voice commands is crucial, especially in multi-user environments. Our proposed solution addresses current system limitations and enhances user privacy. We advocate for equal control measures and data transparency for all users, aligning with the European *General Data Protection Regulation* (GDPR) (Art. 15). This regulation emphasizes individuals' rights to access and control their personal data, underlining the need for equal transparency and control for all device users.

A. Current State

The procedure for configuring a smart speaker is depicted in Fig. 3a. Typically, the device owner is prompted to create or log in to an account with the manufacturer. After setting up the location and WiFi connection, users can optionally enable voice recognition. However, this optional setup does not adhere to the "privacy-by-default" principle outlined in GDPR Art. 25, potentially leaving users unaware of its implications. Additionally, traditional systems rely solely on a single wake word for identification and authentication, making them vulnerable to replay attacks. Our solution counters this by authenticating the entire voice command, effectively mitigating the risk of unauthorized access through replayed wake words [19, 24].

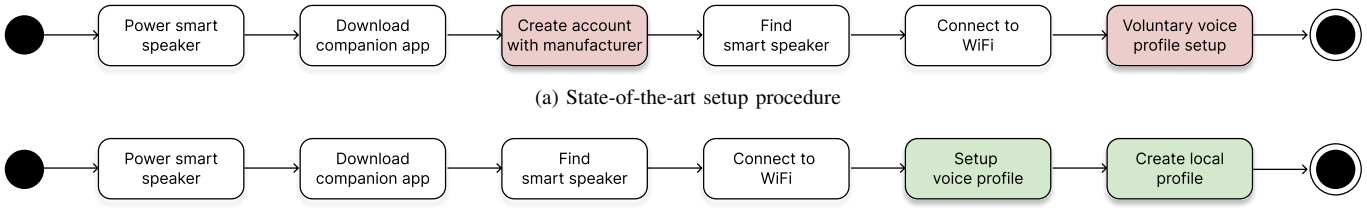
B. Proposed Solution

For new users engaging with the smart speaker, the setup process is illustrated in Fig. 3b. If unrecognized, users are prompted to create a local account linked to their voice profile, securely stored on the device using advanced encryption algorithms. Continuous authentication occurs throughout the entire voice command, not just the wake word, based on this profile. All voice interactions and transcripts remain local, stored exclusively in dedicated folders on the smart speaker for authenticated users. No recordings or transcripts are stored on remote servers, and recordings by unknown speakers are not retained. Fig. 4 outlines the user interaction process and history retrieval method. During initial setup, users use a companion app to share WiFi credentials and initial audio recordings, generating embeddings for the local profile. Subsequently, issued voice commands undergo embedding comparison with existing local profiles. If a match is found, the interaction is stored locally; otherwise, it's discarded. Audio is forwarded to AVS for processing only after verification by our voice check component. Through our companion app, users can access their locally stored past voice interactions securely.

IV. Proof-of-concept Implementation

A. Data Preprocessing for Input

To create speaker embeddings and local profiles, essential for our system's functioning, we utilize a specialized model. Before training this model, we preprocess the audio data. This involves converting audio files to a uniform sample rate and mono channel, resampling to 16 kHz, and extracting 40 MFCCs from each waveform, a common technique in voice recognition systems [25]. These coefficients capture unique audio characteristics crucial for speaker recognition. During model training, we use triplets of audio samples: an anchor, as well as a positive sample from the same speaker, and a negative sample from a different speaker. These triplets are trimmed to ensure uniformity and then used for MFCC feature extraction. We trained our model using the "LibriSpeech ASR Corpus" training set, consisting of 360 hours of clean speech, and evaluated its performance on the test set containing 5.4 hours of speech data (see Tab. I). This initial performance check was conducted within the broader training phase.



(a) State-of-the-art setup procedure

(b) Proposed setup procedure

Fig. 3. Comparison of the current and our proposed setup and registration process

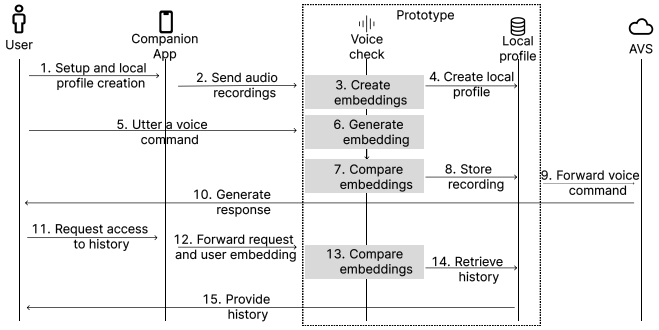


Fig. 4. Sequence diagram for user interactions with our prototype

TABLE I
DATASETS USED FOR TRAINING AND TESTING

Name	Hours	Speakers	Training	Testing	Used in Sec.	Language
LibriSpeech Train [26]	360	921	X		IV-A	English
LibriSpeech Test [26]	5.4	40		X	IV-A	English
Speech Accent Archive [27]	ca. 4.3	709		X	V-A	English
Own collected dataset	0.9	17		X	V-E	German

B. Neural Network Architecture

To implement our continuous authentication framework, we chose a DNN, specifically utilizing LSTM networks within the PyTorch framework. The three-layer LSTM network architecture (Fig. 5) processes audio features, particularly MFCCs, with a hidden state size of 64 capturing temporal dependencies. During training, the LSTM network takes batches of audio features with a batch size of eight and a fixed input dimensionality of 40 MFCC features, structured as $[batch\_size, sequence\_length, feature\_dim]$. We chose a bidirectional model for this work due to its superior performance in our experiments. For embedding creation, input dimensions can vary but must align with the network's input dimensionality. The training process employs a triplet loss function (Fig. 6), minimizing the distance between anchor and positive samples while maximizing the distance between anchor and negative samples to distinguish speakers. During the usage of our speaker recognition system, a user's voice characteristics are captured in a speaker embedding, stored as an array on our prototype. To authenticate new audio recordings, we employ the cosine similarity function. This function computes a score indicating the similarity between the audio sample and the stored embedding. A score closer to 1.0 suggests a higher likelihood of both recordings originating from the same speaker. As the cosine similarity produces a numeric value, a threshold is necessary to differentiate between replay attacks and legitimate commands. The selection and discussion of this threshold are addressed in Sec. VI-A.

V. Evaluation Methodology

The evaluation of our solution is multi-fold. It relies on different existing datasets and our own collected dataset for this purpose (see Tab. I). By using them, we analyze the following dimensions: (1) detection of replay attacks (Sec. V-B), (2) efficacy of replay detection for different accents (Sec. V-C), (3) resilience to background noise and muffled speech (Sec. V-D), and (4) application for another language and using specific smart speakers voice commands (see Sec V-E). In addition to the implementation itself, we validate the feasibility of our solution in practice by measuring the incurred overheads: (1) verification duration, (2) CPU usage, and (3) memory usage (see Sec V-F). In this section, we present the applied methodology, before presenting the results in Sec. VI.

A. Testing Data

Note that we are aware of the existence of the "MASSIVE" [28] and "SLURP" [29] datasets, which include real interactions with "Alexa". However, the recordings are often not provided with a wake word and are therefore not suitable for our use case. As no datasets with real wake words and voice commands are available, we have decided to test our model on the "Speech Accent Archive" [30]. This dataset contains recordings of the same English sentence pronounced by speakers with different accents, resulting in ca. 4.3 hours of speech (see Tab. I). We use 579 English native speakers for our initial tests. The recordings were done in a lab environment with professional microphones. The sentence reads as follows:

> *"Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station."*

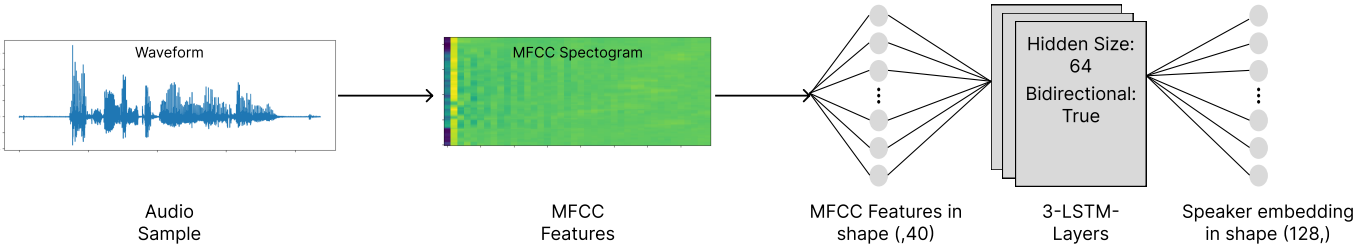


Fig. 5. LSTM network architecture

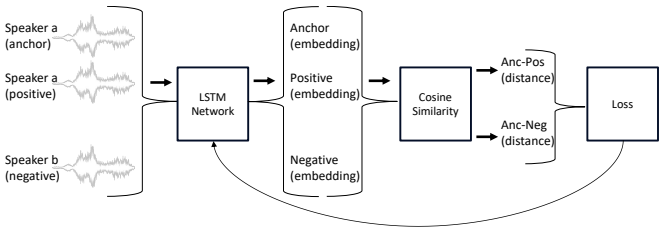


Fig. 6. Steps of the triplet loss function

We segment the recordings by separating the wake word *"Please call Stella."* from the remainder of the statement. We allocate the first 75% of the remaining portion of the sentence to create the speaker embeddings and use the last 25% to represent the rest of the voice command for validation, as illustrated in Fig. 7.

B. Replay Attack Detection

We evaluate various wake word and voice command combinations from different users using the test utterance in Fig. 7. With English native speakers from the "Speech Accent Archive", as mentioned in Sec. V-A, each run involves randomly selecting 20 speaker IDs. Among these, ten have pre-stored speaker embeddings, marking them as legitimate users. Pairing these wake words with the second segment of voice commands from the remaining 20 speakers yields 200 combinations per run. Evaluation metrics including precision, recall, F1-score, accuracy, False Acceptance Rate (FAR), and False Rejection Rate (FRR) assess system performance in distinguishing between replay attacks and legitimate commands.

C. Impact of Accents

We evaluate the impact of accents by testing our approach on speakers with French and German accents, in addition to native English speakers. Using the "Speech Accent Archive" dataset [27], we include 85 French accent and 45 German accent speakers. This broader evaluation ensures the system's adaptability to diverse linguistic environments beyond its initial training on native English speakers.

D. Impact of Background Noise and Muffled Speech

To further test the robustness of our system, we consider both background noise and muffled speech. The types of noise introduced include dog barks, traffic sounds, and rainfall. Using Python and the Librosa library, we have included these noise elements into the individual recordings of 579 English native speakers from "The Speech Accent Archive" dataset [27]. The noise data is either looped or truncated to match the length of each audio clip, to simulate real-world noisy environments. For simulating muffled speech, we have applied a low-pass filter to the recording of the same speakers, with a cutoff frequency set at 1000 Hz. Applying the filter aims at simulating voice commands from users behind a closed door or window.

E. Impact of Command Duration, Language, and Simulated Multi-user Environment with Real-world Participants

The objectives of our user study is to verify if the spoken language and the command duration have an impact on the performance and investigate the performance in a simulated multi-user environment. Approved by our Data Protection Officer, the study involved 17 German-speaking participants. Note that our institution does not have a formal IRB process. However, we took steps to minimize potential harms from our study by adhering to our institution's code of ethics and standards of good scientific practice. Our participants were recruited in the circle of friends, family, and colleagues, as social relationships are not expected to introduce biases. Participants were first asked to complete an online questionnaire to gather their demographics. For capturing their voice commands, we have developed a tool that displays voice commands and wake words to be read aloud. Each participant made recordings for three different wake words, "Alexa", "Ok Google", and "Hey Siri". For each wake word, each participant recorded 10 distinct commands twice: Once with the wake word and once without it, additionally the three wake words have been recorded separately. This resulted in a total of 63 recordings per participant. We have selected these commands based on potential real-world interactions with smart speakers. For example, "What's on my calendar for tomorrow?" and "Turn on the light!". The duration for completing the study was about 25 minutes. Following the user study, we analyzed the dataset and identified key differences compared to existing datasets. Our recordings are shorter (two to four seconds) with occasional silent segments and varying volume levels, resulting in ca. 0.9 hours of total speech (see Tab. I). To standardize the data, we preprocessed the audio by removing silent segments and normalizing volume levels. Subsequently, we categorized the recordings into three groups: (1) wake word only, (2) wake word combined with a voice command, and (3)

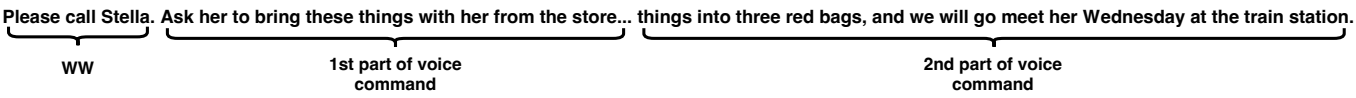


Fig. 7. Visualization on how the utterance of the dataset is separated

voice command only (see Sec.V-B and Sec.IV-A). We simulate a multi-user scenario with up to five participants interacting with the same smart speaker. While the average household size in developed countries is typically two to three people [31, 32], we expanded it to five to account for guests, reflecting real-world conditions. Using a permutation-based approach, we generate embeddings for each participant based on their voice commands, including randomly chosen wake words. Then, we compare all participants using 20 unique utterances for each wake word. This process is repeated 100 times, resulting in 150,000 comparisons, with a threshold set at 85% for user verification (see Sec. VI-A). Speaker embeddings are created from recordings combining wake words and voice commands, compared with voice commands alone.

F. Incurred Overheads

To evaluate performance metrics (latency, CPU, and memory overheads), we implemented our solution on a Raspberry Pi 4B with a ReSpeaker 4-Mic Array for input and a portable 3.5 mm music box for output. The Raspberry Pi OS (64-bit ARM version) was used, along with Python, PyTorch, and Node.js for data processing, model deployment, server operations, and BLE communications. As part of our hardware setup, we developed a companion app using React Native on an iPhone 12 Pro, enabling BLE connection setup, WiFi configuration, embedding creation, and recording access with the Raspberry Pi server. Performance evaluation utilized Python and the psutil library to monitor CPU and memory usage, recording verification times, and AVS response times. Analysis was based on 60 recordings per participant from our user study, focusing on verification times, CPU and memory usage, and latency variations with different recording lengths and times of day. Experiments were conducted hourly from noon to midnight on January 26th, 2024, with minimal recordings sent to AVS, emphasizing local verification on the Raspberry Pi. This setup provided insights into model efficiency and scalability under diverse conditions, using standard statistical methods to assess system performance.

VI. Results

In this section, we present the findings garnered from our series of experiments, which encompass the following key aspects: (1) the efficacy of replay attack detection, (2) the influence of various accents, (3) the impact of background noise and muffled speech, and (4) application for another language and using specific smart speakers voice commands (see Sec V-E). In addition to the implementation itself, we validate the feasibility of our solution in practice by measuring the incurred overheads: (1) verification duration, (2) CPU usage, and (3) memory usage (see Sec V-F).

A. Replay Attack Detection

To find the optimal threshold for detecting both replay attacks and legitimate commands, we conducted 2,000 runs of our detection mechanism across thresholds from 60% to 95%. The results, depicted in Fig. 8, indicate an optimal threshold range between 85% and 86%. With a lower threshold, false positives increase due to replayed recordings being mistaken as legitimate commands. Conversely, higher thresholds lead to more false negatives for legitimate commands. Notably, the recall score for legitimate commands decreases when thresholds exceed 90%. Moreover, detecting replay attacks is more stable than detecting legitimate commands. This might be the case because, in each run of our detection mechanism, we have 200 comparisons, of which 190 are replay attacks and only 10 are legitimate commands. In other words, there is an imbalance in the dataset, with more instances originating from unauthorized users. We have, however, chosen this style of comparison to test and stress our detection system by confronting a single verified speaker embedding with a multitude of different embeddings.

B. Impact of Accents

Results from the same dataset as in Sec. VI-A, but with users having French or German accents, are shown in Fig. 9. English native speakers serve as a baseline for comparison. Using a threshold of 85%, consistent with Sec. VI-A, we observe consistent performance in detecting replay attacks across all accents (f1-score 99%). However, the accuracy in identifying legitimate commands varies with the accent, with the highest performance for French and the lowest for German speakers. Overall, our solution achieves a minimum f1-score of 90%. Tab. II displays FARs and FRRs for legitimate speakers, showing low FARs across all accents, indicating robust security. FRRs are highest for speakers with a German accent.

C. Impact of Background Noises and Muffled Speech

The results in Fig. 9 reveal a decrease in accurate identification of legitimate commands when exposed to traffic and rain noises. This decline may be due to these noises masking vocal features crucial for verification. Conversely, the impact of sporadic dog barks is minimal, likely because they do not consistently overlap with vital vocal patterns. Additionally, simulating muffled speech (as discussed in Sec. V-D) leads to a similar decline in accurate verification, suggesting challenges for malicious users attempting replay attacks using recordings made behind obstacles.

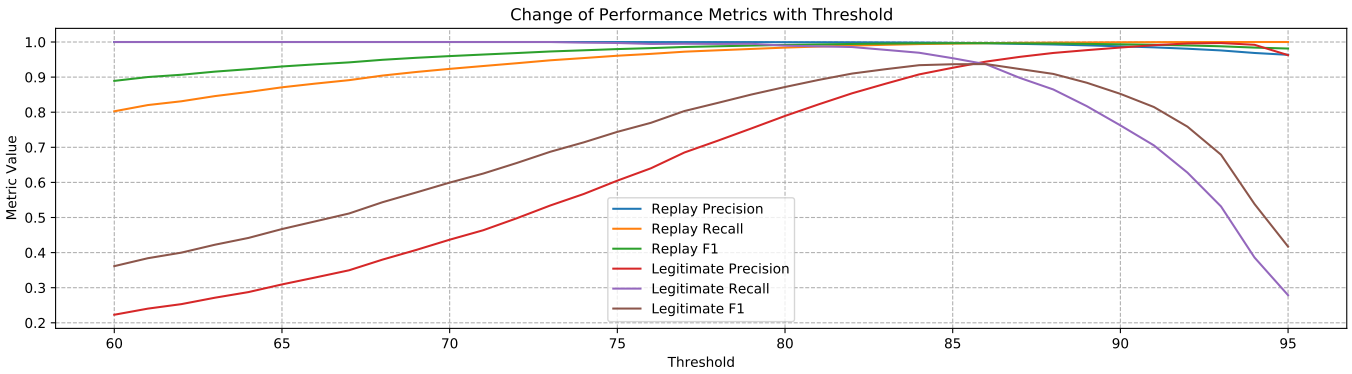


Fig. 8. Change of performance metrics for different threshold over 2000 runs

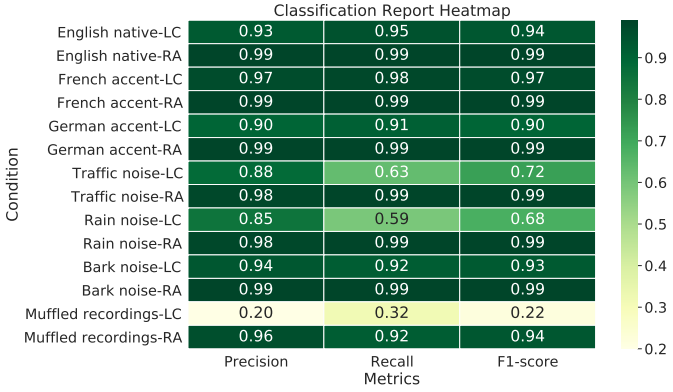


Fig. 9. Classification report for legitimate commands (LC) and replay attacks (RA) for different accents, noise types, and muffled recordings obtained with threshold set to 85%

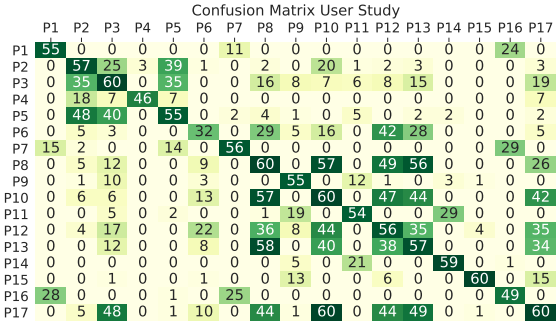


Fig. 10. Confusion matrix of 17 participants with threshold set to 85%

TABLE II
FAR AND FRR VALUES FOR SPEAKERS WITH DIFFERENT ACCENTS.

Accent	FAR	FRR
German Accent	0.57%	8.97%
English Native	0.43%	5.11%
French Accent	0.20%	2.12%

D. Impact of Language, Command Duration, and Simulated Multi-user Environment

Despite our participants using German commands, unlike the English commands in "The Speech Accent Archive" [27], and despite shorter recordings (approximately two to four seconds compared to about 24 seconds), we demonstrate the feasibility of speaker recognition with German commands, albeit with reduced performance compared to English commands. Tab. III displays the results for simulated multi-user environments, as described in Sec.V-E. As expected, performance weakens with more users. While high recall ensures accurate storage of nearly all authenticated interactions, low precision indicates potential access for unverified users to sensitive information or controls. Results for groups of two, three, four, and five speakers are provided in Tab. III. The confusion matrix in Fig. 10 illustrates system performance for our 17 participants'. While the system exhibits high recall for correct identifications, it also shows some misidentifications, leading to lower precision. These errors often correlate with speakers' genders: male voices are more frequently mistaken for other male voices, and likewise for female voices.

TABLE III
PERFORMANCE METRICS FOR DIFFERENT NUMBERS OF SPEAKERS.

Speakers	Accuracy	Precision	Recall	F1 Score
2 Speakers	0.89	0.94	0.88	0.90
3 Speakers	0.89	0.83	0.90	0.85
4 Speakers	0.89	0.77	0.90	0.82
5 Speakers	0.89	0.69	0.90	0.77

E. Verification and Communication Time, CPU and Memory Usage

Tab. IV presents key performance metrics and system resource usage obtained from the prototype described in Sec. V-F, based on data from the 17 participants in our study. Each participant contributed 60 utterances for testing. The initial higher standard deviation in verification time for the first speaker (489 ms) is likely due to the computational overhead of initializing the PyTorch model and its libraries. Despite variations across speakers possibly influenced by individual characteristics, accents, or background noise, the consistent patterns in verification times and resource usage demonstrate the reliability of our approach. To validate the consistency of our previously obtained results, we replicated

TABLE IV
PERFORMANCE METRICS FOR SPEAKER VERIFICATION AND SYSTEM RESOURCE USAGE: AVT (AVERAGE VERIFICATION TIME), SDVT (STANDARD DEVIATION OF VERIFICATION TIME), ACU (AVERAGE CPU USAGE), SDCU (STANDARD DEVIATION OF CPU USAGE), AMU (AVERAGE MEMORY USAGE), SDMU (STANDARD DEVIATION OF MEMORY USAGE)

Speaker	AVT (ms)	SDVT (ms)	ACU (%)	SDCU (%)	AMU (%)	SDMU (%)
1	226	489	49.24	3.12	21.54	0.192
2	175	22	48.96	1.03	21.88	0.057
3	164	9	49.27	1.11	21.61	0.393
4	184	22	48.45	1.12	21.29	0.125
5	176	16	48.80	1.15	21.43	0.045
6	190	82	49.07	2.60	21.91	0.207
7	175	19	48.22	1.38	21.85	0.200
8	175	17	48.39	1.51	21.69	0.031
9	170	13	48.89	0.91	21.67	0.043
10	208	36	48.07	1.18	22.19	0.253
11	189	20	48.29	1.08	22.50	0.042
12	180	18	48.43	1.09	22.49	0.026
13	242	144	51.40	6.93	23.14	0.320
14	213	100	50.54	5.78	23.34	0.069
15	218	141	50.48	5.46	23.33	0.092
16	211	124	51.41	6.47	22.99	0.306
17	213	134	50.72	5.64	23.21	0.201
Avg:	**195**	**83**	**49.33**	**2.80**	**22.24**	**0.153**

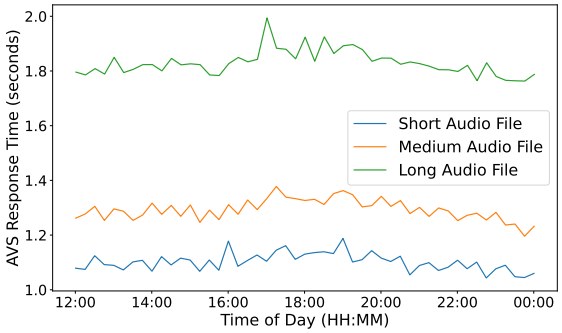


Fig. 11. AVS (Amazon Voice Service) response times

TABLE V
AVERAGE AVS RESPONSE TIMES (ARP) AND STANDARD DEVIATION OF RESPONSE TIME (SRT) FOR DIFFERENT AUDIO FILE LENGTHS

Audio File Length	ARP (s)	SRT (ms)
Short (about 2 seconds)	1.10	32
Medium (about 4 seconds)	1.29	37
Long (about 8 seconds)	1.83	46

the experiments at various times of the day to account for potential variations in AVS availability. Fig. 11 illustrates the average response times recorded on January 26th, 2024, from noon to midnight. It is important to note that both CPU and memory usage are independent of AVS availability and, as such, were not included in this evaluation. We tested three distinct lengths of audio recordings as detailed in Sec. V-F: short (about 2 s), medium (about 4 s), and long (about 8 s). The standard deviations observed were 32 milliseconds for short audio recordings, 37 milliseconds for medium ones, and 46 milliseconds for long ones, confirming that response time differences remain marginal throughout different times of the day (see Tab.V). The results indicate that response time is proportionate to the length of the voice recordings. In conclusion, our proof-of-concept implementation and its thorough evaluation underscore the viability of our approach in real-world conditions.

VII. SUMMARY, DISCUSSION, AND FUTURE WORK

Considering all findings, we have demonstrated successful speaker authentication beyond the wake word. Unlike current text-dependent systems relying solely on a fixed wake word, our proposed text-independent and offline approach is versatile. Even English speakers with various accents can be authenticated, leveraging a neural network trained solely on native English speakers' utterances. Though recordings with added background noise reduce verification performance, additional preprocessing to filter out noise could yield comparable results to clean recordings. Muffled recordings, simulating closed-door scenarios, are not verified by our system, making it harder to trick authentication. We further evaluated our authentication scheme using real-world voice commands from a user study, testing its robustness with speakers of different languages. With an f1-score of 77%, our results suggest room for improvement. The shorter duration of recordings (2–4 seconds) and the language difference contribute to the suboptimal performance. Additionally, our local verification introduces a slight delay to AVS interactions. However, considering average response times, this delay is not expected to significantly impact the user experience. Despite its effectiveness against honest-but-curious users and wake word impersonation, our system still faces limitations. Replay attacks using full command recordings remain possible. Additionally, advancements in voice cloning pose a potential threat [33]. Studying its impact on our authentication scheme and assessing our system's resilience to such attacks would be valuable. Finally, users may have varying preferences regarding data sharing and

sensitivity. Hence, allowing users to control the accessibility of information, especially to guests or unauthenticated individuals, is crucial. Exploring privacy configuration interfaces tailored to these needs is an area for future research.

VIII. Conclusion

We have developed a speaker recognition system focused on detecting replayed wake word attacks and providing continuous authentication during voice commands. Through experiments with three datasets, we optimized f1-score, FAR, and FRR. Our system achieves high precision, recall, f1-score, and accuracy rates of 93%, 95%, 94%, and 99%, respectively, for authenticating native English speakers. Furthermore, we report a FAR of 0.43% and a FRR of 5.11%. Overall, considering the response delays observed for short, medium, and long audio files, the additional delay introduced by our local verification system does not significantly impact the overall user experience. Completing our continuous authentication approach, we propose the implementation of a local profile-based voice authentication platform as an initial filtering layer that could be integrated ahead of the core voice assistant ecosystem. Using it, the mandatory online account registration with device manufacturers would be prevented. By storing and managing user data locally on the smart speaker, our proposed system hence enhances user privacy and control over their data, particularly in multi-user settings.

References

[1] Y. Wang, W. Cai, T. Gu, W. Shao, Y. Li, and Y. Yu, "Secure Your Voice: An Oral Airflow-Based Continuous Liveness Detection for Voice Assistants," *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2019.

[2] S. Pradhan, W. Sun, G. Baig, and L. Qiu, "Combating Replay Attacks Against Voice Assistants," *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2019.

[3] Y. Meng, J. Li, H. Zhu, Y. Tian, and J. Chen, "Privacy-Preserving Liveness Detection for Securing Smart Voice Interfaces," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[4] M. Sahidullah, D. A. L. Thomsen, R. G. Hautamäki, T. Kinnunen, Z.-H. Tan, R. Parts, and M. Pitkänen, "Robust Voice Liveness Detection and Speaker Verification Using Throat Microphones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[5] J. Shang and J. Wu, "Voice Liveness Detection for Voice Assistants Using Ear Canal Pressure," in *Proc. of the 7th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020.

[6] Q. Yang, K. Cui, and Y. Zheng, "VoShield: Voice Liveness Detection with Sound Field Dynamics," in *Proc. of the 42nd International Conference on Computer Communications (IEEE INFOCOM)*, 2023.

[7] Y. Lee, Y. Zhao, J. Zeng, K. Lee, N. Zhang, F. H. Shezan, Y. Tian, K. Chen, and X. Wang, "Using Sonar for Liveness Detection to Protect Smart Speakers Against Remote Attackers," *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2020.

[8] L. Hernández Acosta and D. Reinhardt, "A Survey on Privacy Issues and Solutions for Voice-Controlled Digital Assistants," *Pervasive and Mobile Computing (PMC)*, 2021.

[9] J. L. Kröger, O. H.-M. Lutz, and P. Raschke, "Privacy Implications of Voice and Speech Analysis–Information Disclosure by Inference," in *Proc. of the 14th IFIP Int. Summer School on Privacy and Identity Management (IFIP SC)*, 2019.

[10] Mozilla. (2021) Amazon Echo Dot. [Online]. https://foundation.mozilla.org/en/privacynotincluded/amazon-echo-dot/.

[11] ——. (2021) Google Nest Mini. [Online]. https://foundation.mozilla.org/en/privacynotincluded/google-nest-mini/.

[12] K. Sharif and B. Tenbergen, "Smart Home Voice Assistants: A Literature Survey of User Privacy and Security Vulnerabilities," *Complex Systems Informatics and Modeling Quarterly (CSIMQ)*, 2020.

[13] M. Vimalkumar, S. K. Sharma, J. B. Singh, and Y. K. Dwivedi, "'Okay Google, What About My Privacy?': User's Privacy Perceptions and Acceptance of Voice Based Digital Assistants," *Computers in Human Behavior*, 2021.

[14] P. Cheng and U. Roedig, "Personal Voice Assistant Security and Privacy—a Survey," *Proc. of the IEEE*, 2022.

[15] N. Malkin, J. Deatrick, A. Tong, P. Wijesekera, S. Egelman, and D. Wagner, "Privacy Attitudes of Smart Speaker Users," *Proc. on Privacy Enhancing Technologies (PoPETs)*, 2019.

[16] F. Brasser, T. Frassetto, K. Riedhammer, A.-R. Sadeghi, T. Schneider, and C. Weinert, "VoiceGuard: Secure and Private Speech Processing," in *Proc. of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.

[17] Rhasspy. (2019) Rhasspy Voice Assistant. [Online]. https://rhasspy.readthedocs.io/en/latest/.

[18] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril *et al.*, "Snips Voice Platform: An Embedded Spoken Language Understanding System for Private-by-Design Voice Interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[19] R. He, X. Ji, X. Li, Y. Cheng, and W. Xu, ""OK, Siri" or "Hey, Google": Evaluating Voiceprint Distinctiveness Via Content-based PROLE Score," in *Proc. of the 31th USENIX Security Symposium*, 2022.

[20] C. Zhang and K. Koishida, "End-To-End Text-Independent Speaker Verification With Triplet Loss on Short Utterances," in *Proc. of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017.

[21] Z. Bai and X.-L. Zhang, "Speaker Recognition Based on Deep Learning: An Overview," *Neural Networks*, 2021.

[22] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized End-To-End Loss for Speaker Verification," in *Proc. of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[23] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker Diarization With LSTM," in *Proc. of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[24] M. Combs, C. Hazelwood, and R. Joyce, "Are You Listening?–an Observational Wake Word Privacy Study," *Organizational Cybersecurity Journal: Practice, Process and People*, 2022.

[25] Q. Wang. (2023) Speaker recognition from scratch. [Online]. https://github.com/wq2012/SpeakerRecognitionFromScratch.

[26] V. Panayotov. (2015) LibriSpeech: An ASR Corpus based on Public Domain Audio Books. George Mason University. [Online]. https://www.openslr.org/12.

[27] S. H. Weinberger and S. A. Kunath, "The Speech Accent Archive: Towards a Typology of English Accents," in *Corpus-Based Studies in Language Use, Language Learning, and Language Documentation*, 2011.

[28] J. FitzGerald, C. Hench, C. Peris, S. Mackie, K. Rottmann, A. Sanchez, A. Nash, L. Urbach, V. Kakarala, R. Singh, S. Ranganath, L. Crist, M. Britan, W. Leeuwis, G. Tur, and P. Natarajan, "MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[29] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A Spoken Language Understanding Resource Package," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[30] S. Weinberger. (2015) Speech Accent Archive. George Mason University. [Online]. http://accent.gmu.edu.

[31] N. McCarthy. (2019) German Households Are Getting Smaller. Statista. [Online]. https://www.statista.com/chart/18820/average-number-of-household-members-in-german-federal-states/.

[32] Statista Research Department. (2023) Average Number of People per Household in the United States From 1960 to 2022. Statista. [Online]. https://www.statista.com/statistics/183648/average-size-of-households-in-the-us/.

[33] P. Neekhara, S. Hussain, S. Dubnov, F. Koushanfar, and J. McAuley, "Expressive Neural Voice Cloning," in *Proc. of the 13th Asian Conference on Machine Learning (ACML)*, 2021.